Deniable RSA Signature
The Raise and Fall of Ali Baba

Serge Vaudenay

EPFL
CH-1015 Lausanne, Switzerland
http://lasecwww.epfl.ch

Abstract. The 40 thieves realize that the fortune in their cave is vanishing. A rumor says that Ali Baba has been granted access (in the form of a certificate) to the cave but they need evidence to get justice from the Caliph. On the other hand, Ali Baba wants to be able to securely access to the cave without leaking any evidence. A similar scenario holds in the biometric passport application: Ali Baba wants to be able to prove his identity securely but do not want to leak any transferable evidence of, say, his date of birth.
In this paper we discuss the notion of offline non-transferable authentication protocol (ONTAP). We review a construction based on the GQ protocol which could accommodate authentication based on any standard RSA certificate. We also discuss on the fragility of this deniability property with respect to set up assumptions. Namely, if tamper resistance exist, any ONTAP protocol in the standard model collapses.

1 Prolog: Supporting Ali Baba's Crime in a Fair Way

Many centuries after the 1001 nights, Queen Scheherazade revisits her story of Ali Baba [23]: Ali Baba is well known to be granted access to the cave of the forty thieves. However, any proof of that could be presented to the Caliph would let the king no other choice than condemn him for violating the thieves' property. So far, the thieves did not succeed to get such evidence. Actually, Ali Baba's grant to open the cave has the form of an RSA certificate (signed by Scheherazade, the authority in this story) assessing that Ali Baba is authorized to open the cave. As the cave recognizes the queen's signature, it opens for Ali Baba. However, this certificate can constitute some evidence to convict Ali Baba and Scheherazade. Otherwise, the existence of this certificate can just be denied and nobody would even risk to implicitly accuse the Queen without any evidence. Since, the thieves could play some active attack (sometimes also called thief-in-the-middle attack) between the cave and Ali Baba, the access control protocol must be such that, while being secure so that nobody without any valid certificate could open the cave, the thieves have no way to get any transferable proof which could convict Ali Baba. Indeed, Ali Baba is running a protocol with the cave, proving possession of the RSA signature but in a deniable way. In this paper we describe this protocol which is based on the Guillou-Quisquater (GQ) protocol [14,15].

History of this protocol. This problem appeared with the application of the biometric passport [1]. In this application, the passport holds a signature (by government authorities) assessing the identity of the passport holder. The identity is defined by a facial picture, a name, a date of birth, a passport number, and its expiration date. One problem with this application (called "passive authentication") is that any passport reader (or any reader getting through the access control protocol) can get this signature which can later be collected or posted for whatever reason. This would raise privacy concerns. Some closely related protocols such as [2] were proposed for slightly different applications, based on ElGamal signatures. At Asiacrypt 2005, after [2] was presented, Marc Girault suggested that the GQ protocol [14,15] could be used to prove knowledge of an RSA signature in a zero-knowledge (ZK) way. The basic GQ protocol is not zero-knowledge though, so we have to enrich it. The application

to the biometric passport was suggested by Monnerat, Vaudenay, and Vuagnoux in [20,28,30]. At ACNS 2009, Monnerat, Pasini, and Vaudenay [17] presented this enriched protocol together with a proof of security. The protocol is called an *offline non-transferable authentication protocol (ONTAP)*. We review this result in this paper.

Our ONTAP protocol involves three participants: the authority (Queen Scheherazade), the holder (Ali Baba), and the server (the cave). The authority is trusted. It holds a secret key for signature but the other participants do not hold any secret, a priori. However, the protocol should be protected against a cheating prover trying to open the cave, a cheating verifier trying to collect (offline) transferable evidence from Ali Baba. Non-transferability was introduced in [6,16]. We distinguish here offline evidence (proofs which could be shown to a trial) from online evidence (proofs involving some action by the judge during the protocol) because we do not assume the Caliph to be willing to participate to some online attack. Although weaker than online non-transferability, offline non-transferability is implied by regular zero-knowledge. More precisely, it is implied by deniable zero-knowledge [22]. The advantage is that it can be achieved without deploying a PKI for verifiers. Although our framework could accommodate any type of standard signature, we focus on RSA signatures which require the GQ protocol. (Signatures based on ElGamal would require the Schnorr protocol [24,25] instead.)

Deniability is a fragile notion. Indeed, we often prove deniability in zero-knowledge protocols by the ability to simulate the transcript by rewinding the verifier. This implicitly assumes that any computing device could be rewinded. In practice, there are many hardware systems which are assumed not to be rewindable. Namely, tamper-proof devices are not rewindable. This implies that we could loose deniability by implementing a verifier in such a device as shown by Mateus and Vaudenay [18,19]. We conclude this paper by telling how Ali Baba was caught in this way.

Related notions. Several notions similar to ONTAP exist but none of them fully match our needs. *Non-transitive signatures* [10,21] and *deniable authentication* [11] only involve two participants (which would imply that Ali Baba must know the authority secret key, which does not fit our application). *Invisible signatures* (a.k.a. undeniable signatures) [8] also involve two participants and do not always accommodate non-transferable properties, which is one of our main requirements. *Designated confirmer signatures* [7] involve three participants. These extend invisible signatures by protecting the verifier from signers unwilling to participate in the protocol. A typical protocol would be some unreliable signer delegating a trusted confirmer to participate in the proof protocol. In our scenario, the signer (Scheherazade) is trusted but the confirmer (Ali Baba) may be not. *Universal designated-verifier signatures (UDVS)* [27] involve three participants as well, but rely on a PKI for verifiers. In our scenario, we do not want to deploy a new PKI for the cave. A weaker notion is the *universal designated verifier signature proof (UDVSP)* [2]. The difference with our scenario is that the verifier in the protocol is assumed to be honest. There are also stronger notions such as *credential ownership proofs (COP)* [26], but they are more involved than our solution and do not always fit standard signatures.

2 Log: Making ONTAP from Standard Signature Schemes

2.1 Zero-Knowledge Proof based on GQ

In the literature, there have been several definitions for Σ-protocols. (See [4,9].) For our purpose, we change a bit the definition.

Definition 1. *Let R be a relation which holds on pairs (x,w) in which $x \in D_x$ is called an instance and $w \in D_w$ is called a witness. Let κ be a function mapping x to a real number. A Σ-protocol for*

R is defined by one interactive algorithms P, some sets $D_a(x)$, $D_r(x)$, $D_z(x)$, and some verification algorithm Ver. *If $(x,w) \in R$, ϖ is a random tape, and r is a random element of E, we denote $a = P(x,w;\varpi)$, $z = P(x,w,r;\varpi)$, and $b = \mathsf{Ver}(x,a,r,z)$. Actually, we define an interactive algorithm V by $V(x,a;r) = r$ and $V(x,a,z;r) = \mathsf{Ver}(x,a,r,z)$ as the final output of the protocol. So, it is a 3-move protocol with transcript (a,r,z), common input x, and output b. The protocol is a κ-weak Σ-protocol if there exists some extra algorithms* Sim *and* Ext *such that the following conditions are satisfied.*

- *(efficiency) algorithms P,* Ver, Sim, *and* Ext *are polynomially computable (in terms of the size of x)*
- *(uniqueness of response) there exists a function* Resp *such that*

$$\forall x \in D_x \quad \forall a \in D_a \quad \forall r \in D_r \quad \forall z \in D_z \qquad \mathsf{Ver}(x,a,r,z) = 1 \Longleftrightarrow z = \mathsf{Resp}(x,a,r)$$

- *(completeness) we have*

$$\forall (x,w) \in R \quad \forall r \in D_r \quad \forall \varpi \quad P(x,w,r;\varpi) = \mathsf{Resp}(x, P(x,w;\varpi), r)$$

- *(κ-weak special soundness) we have*

$$\forall x \in D_x \quad \forall a \in D_a \quad \forall r \in D_r \quad \Pr_{r' \in E}[(x, \mathsf{Ext}(x,a,r,r', \mathsf{Resp}(x,a,r), \mathsf{Resp}(x,a,r'))) \in R] \geq 1 - \kappa(x)$$

- *(special HVZK) for any $(x,w) \in R$ and a random $r \in D_r$, the distribution of $(a,r,z) = \mathsf{Sim}(x,r)$ and (a,r,z) defined by $a = P(x,w;\varpi)$ and $z = P(x,w,r;\varpi)$ are identical.*

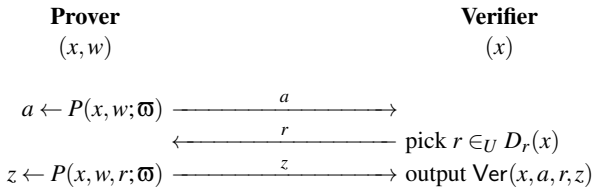

Fig. 1. Σ-Protocol

The main change from the original definitions lies in the introduction of the uniqueness of response and the κ-weak special soundness property. We need these change to make the GQ protocol fit to this protocol model. Indeed, the GQ protocol works as follows. The domain D_x is the set of all triplet $x = (N,e,X)$ where (N,e) is an RSA public key and X is a ciphertext. Then a witness is a decryption, i.e. $((N,e,X),w) \in R \Longleftrightarrow w^e \bmod N = X$. The domains $D_a(x)$ and $D_z(x)$ are $\mathbf{Z}_N$. The domain $D_r(x)$ is the set $\{0,1,\ldots,2^{t(|x|)}-1\}$ for some polynomially bounded function t. The $a = P(x,w;\varpi)$ algorithm first extracts some random $y \in \mathbf{Z}_N^*$ from ϖ, then compute $a = y^e \bmod N$. The $z = P(x,w,r;\varpi)$ algorithm computes $z = yw^r \bmod N$. The Ver algorithm checks that $0 \leq z < N$ and that $z^e \equiv aX^r \pmod N$. We can easily prove [17] that this is actually a κ-weak Σ-protocol for $\kappa(x) = \lceil \frac{2^{t(|x|}}{e} \rceil 2^{-t(|x|)}$ when e is prime. (In practice we use $e = 3$ or $e = 65\,537$ which are both prime.)

In our approach we tweaked the definition of Σ-protocol to make the GQ protocol fit this notion and have a generic construction based on any weak Σ-protocol. Another analysis, dedicated to the GQ protocol, was done by Bellare and Palacio [3].

Σ-protocols are the initial step to build zero-knowledge protocols. These are not completely zero-knowledge because a malicious adversary could cheat by making the challenge be the result of applying the commitment value a through a one-way function. The resulting transcript would be most likely non-simulatable without knowing the secret. This way to cheat is actually used to transform Σ-protocols into digital signature schemes, following the Fiat-Shamir paradigm [12]. This property is terrible for us because a malicious verifier could cheat and collect evidence that the protocol was executed. We seek for the *deniability* property of zero-knowledge. Some extensions of zero-knowledge are defined using stronger set up assumptions such as the random oracle model or the common reference string model are not always deniable, but zero-knowledge in the standard model is always deniable. Deniability was studied by Pass [22].

We use the following classical transform [13] of Σ-protocols into zero-knowledge ones using 5 rounds. We add a commitment stage using a *trapdoor commitment scheme* [5]. This scheme is defined by three algorithms gen, com, and equiv and are such that for $(k,\omega) = \mathsf{gen}(;R)$ and $\mathsf{equiv}(\omega,m,c) = d$ we have $\mathsf{com}(k,m;d) = c$. Here, d is used to open a commitment c on m. ω is used to cheat with the commitment. R is used to verify that the (k,ω) pair is well formed. The transform works as follows.

1. the prover uses some free part of ϖ to generate R, creates a key pair $(k,\omega) = \mathsf{gen}(R)$ for the commitment, and sends k to the verifier
2. the verifier generates r and some decommit value d from its random tape and compute $c = \mathsf{com}(k,r;d)$, then sends c to the prover
3. the prover runs $a = P(x,w)$ using some independent part of ϖ and sends it to the verifier
4. the verifier releases r and d to the prover
5. the prover checks that $c = \mathsf{com}(k,r;d)$ (if $c \neq \mathsf{com}(k,r;d)$, the prover aborts), computes $z = P(x,w,r)$, and sends z and R to the verifier
6. the verifier checks that k is the first output of $\mathsf{gen}(R)$ and that $\mathsf{Ver}(x,a,r,z) = 1$, and answers 1 if and only if both hold

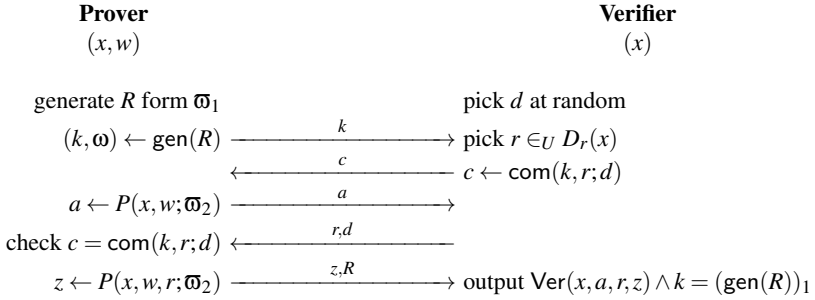

Fig. 2. Generic ZK Construction

This protocol can be shown to be zero-knowledge.

Theorem 2 (Monnerat-Pasini-Vaudenay 2009 [17]). *If* $(P,D_a,D_r,D_z,\mathsf{Ver})$ *is a* κ*-weak* Σ*-protocol and* $(\mathsf{gen},\mathsf{com})$ *is a secure trapdoor commitment scheme, then the above protocol is a zero-knowledge proof of knowledge with soundness error*

$$\kappa'(x) = \max(\kappa(x), 1/\mathsf{Poly}(|x|))$$

for any polynomial Poly.

That is, there is a polynomial-time extractor which, when plugged to a malicious prover passing the protocol with probability $\varepsilon(x) > \kappa'(x)$, can compute a witness for x in time $\mathsf{Poly}(|x|)/(\varepsilon(x) - \kappa'(x))$.

A construction in 4 rounds was proposed in [9] based on the OR proof. However, it is not generic.

2.2 ONTAP Protocols and Application to the Biometric Passport

We now define the ONTAP protocol which can be used to authenticate a message m by means of a certificate (X, w). There are three participants: the signer, the signature holder (prover), and the verifier.

Definition 3. *An offline non-transferable authentication protocol (ONTAP) consists of the following probabilistic polynomial-time (ppt) algorithms.*

- $(K_p, K_s) = \mathsf{setup}(1^\lambda)$ *a key setup algorithm to generate a public key K_p and a private key K_s from a security parameter λ and some random coins*
- $(X, w) = \mathsf{sign}(K_s, m)$ *a signature algorithm to generate from the secret key, a message m and some random coins a signature with a public part X and a private part w*
- *an interactive proof protocol $(\mathcal{P}, \mathcal{V})$ taking (K_p, m, X) as common input, with private input w, and producing a bit b as output.*

The final bit must always be 1 if the algorithms and protocols are well executed. Furthermore, they must provide the security properties of κ'-unforgeability and offline non-transferability as defined by the following games.

In the κ'-unforgeability game, an adversary plays with a challenger. The challenger first generates K_p and K_s and releases K_p. The adversary can then make some signing queries to the challenger who will return the complete signature (X and w). Then, the adversary proposes a message m which was not queried for signature and run the interactive protocol with the challenger by playing himself the role of the prover. We say that the protocol is unforgeable if for any ppt adversary the output bit is 1 with probability at most κ'.

In the offline non-transferable game, a process (either the adversary or a simulator) plays with a challenger. The challenger first generates K_p and K_s and releases K_p. The process can then make some signing queries to the challenger who will return the complete signature (X and w). When the process is the adversary, he can then submit some message m which is signed by the challenger but only the X part of the signature is returned. Then, the adversary can run the interactive protocol with the challenger playing the role of the prover. When the process is the simulator, the submission of m and the protocol execution are skipped. Finally, the process yields the list of all sign queries together with an output string. We say that the protocol is offline non-transferable if for any ppt adversary there exists a ppt simulator such that running the game with both processes generate outputs which are computationally indistinguishable.

Given a traditional digital signature scheme which satisfies the some special properties, we construct an ONTAP protocol based on our generic ZK protocol. First, we need the signature to be splittable into two parts X and w. The X part shall be forgeable without the secret key. Finally, there shall be a weak Σ protocol for which w is a witness for (K_p, m, X). Note that all digital signature schemes satisfy the first two conditions by just setting X to the empty string. The last condition (provability) may however be more efficient by enlarging a simulatable part X as much as we can. The Σ-protocol is enriched into a ZK protocol as on Fig. 2 and we obtain an ONTAP protocol.

Theorem 4 (Monnerat-Pasini-Vaudenay 2009 [17]). *Let $(\mathsf{setup}, \mathsf{sign}, \mathsf{verify})$ be a digital signature scheme such that*

– *the output of* sign *can be written* (X,w)*;*
– *there exists a ppt algorithm* sim *such that for any execution of* $\mathsf{setup}(1^\lambda) = (K_p, K_s)$*, the execution of* $\mathsf{sign}(K_s, m) = (X,w)$ *and of* $\mathsf{sim}(K_p, m) = X$ *generate X's with computationally indistinguishable distributions;*
– *there exists a* κ*-weak* Σ*-protocol for the relation*

$$R((K_p, m, X), w) \Longleftrightarrow \mathsf{verify}(K_p, m, X, w)$$

– *the signature scheme resists existential forgery under chosen message attacks.*

Then, $(\mathsf{setup}, \mathsf{sign})$ *with the protocol from Fig. 3 is an ONTAP scheme which is* κ'*-unforgeable and offline non-transferable where* $\kappa'(\lambda) = \max(\kappa(\lambda), 1/\mathsf{Poly}(\lambda))$*. This holds for any polynomial* Poly*.*

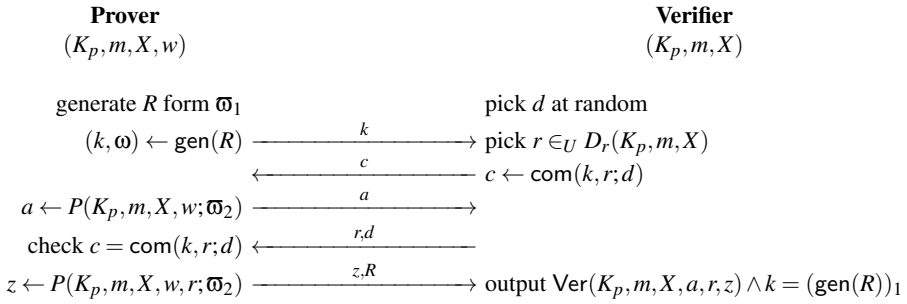

Fig. 3. Generic ONTAP Construction

In practice, the use of a trapdoor commitment is a bit artificial. Indeed, neither the secret key ω nor the equiv algorithms are used. Their only purpose is to write down a formal proof for the protocol to be zero-knowledge. In practice, one may favor the use of a simple commitment based on a hash function. This could be formalized in the random oracle model (ROM) and the specific notion of deniable zero-knowledge could be proven. (See [17].)

For instance, a generic RSA signature for a message m has a public key (N,e) and a secret key d. Here, X is a formatted string based on m and N only and $w = X^d \bmod N$. Checking a signature consists of verifying some predicate $\mathsf{format}(N, m, w^e \bmod N)$ to check that $X = w^e \bmod N$ is a valid formatted string for m. This can be proven by the GQ protocol with the extra input X. Assuming that the RSA signature resists existential forgery under chosen message attacks, we obtain the ONTAP scheme on Fig. 4 with soundness error $\kappa' = \left\lceil \frac{2^t}{e} \right\rceil 2^{-t}$.

In the case of the biometric passport, the passport would play the role of the prover and the reader would be the verifier. So, the passport would prove to the reader that it owns a valid witness w for the formatted message X without leaking any transferable evidence.

By using $e = 65537$ and $t = 16$, we obtain $\kappa' = 2^{-16}$. The workload of the prover consists in computing two exponentials with a 16-bit exponent, one of them being e. Let say this roughly costs 40 multiplications on average using standard square-and-multiply algorithms. An online security of 2^{-16} is pretty good but maybe some people will not be happy with it. If κ' is believed to be too large, we can still execute two instances of the GQ protocol in parallel and obtain $\kappa' = 2^{-32}$ using four exponentials.

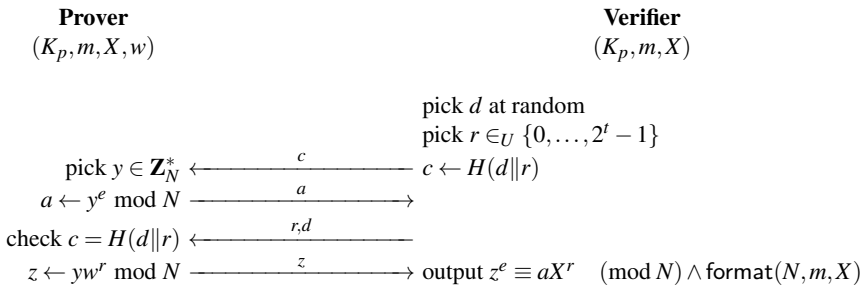

Fig. 4. RSA-Based ONTAP Construction in ROM

By using $e = 3$, a single run of GQ with $t = 1$ leads to $\kappa' = \frac{1}{2}$ at a cost of 2.5 modular multiplications on average. By iterating 16 times we get $\kappa' = 2^{-16}$ and 40 multiplications again. So, the cost with same κ' is roughly the same: we need "2.5 multiplications per bit of online security".

We could also use DSA or any variant such as ECDSA, but with the Schnorr Σ-protocol [24,25] instead of GQ. (See [17].)

3 Epilog: Ali Baba's Trial and the Fall of Deniability

Deniability is a pretty fragile property though. In a world where computability is fully described by regular Turing machines, deniability works. However, some special tamper proof devices may be sealed by the Caliph. In this model, deniability might collapse as shown in [18,19]. What happened is that the thieves used devices sealed by the Caliph.

Indeed, the Caliph provides programmable sealed devices which can be configured by users using any software but with the property that the hash of the loaded software is permanently displayed by the device. That is, the device owner could load it with its favorite software s and the device would always display communications from s attached to $H(s)$. So, another user could be convinced that the displayed message is the result of executing s on a trusted platform. Those devices can, for instance be used in payment terminals where the vendor loads some open software and invite customers to type their credential. Customers can check that the device is genuine and running the publicly certified software.

Unfortunately, the existence of these devices changes the set up assumptions in the computational model. Namely, there are now devices executing some program which cannot be interrupted or rewinded. These devices could also be used maliciously to break the ONTAP protocol. Indeed, the 40 thieves can load a trusted device with the code s of the verifier and display its view. Then, showing the device after the protocol succeeds would prove that a trusted hardware has executed the verifier code in a honest way and seen the displayed accepting view which includes the instance x. Then, the thieves impersonate the cave, replay messages to and from the device, and just wait until Ali Baba wants to enter the cave. After Ali Baba executes the protocol, the device ends up by displaying a reference x to Ali Baba owing a certificate in an accepting view, showing evidence that a prover successfully proved possession of a witness for x. The 40 thieves can then go to the Caliph and show the device as a proof that there exists a valid certificate for x: that Ali Baba is granted access to the cave. This convicts Ali Baba (and Scheherazade as well, which is embarrassing for the Caliph).

One can easily realize that no ONTAP protocol in the standard model can survive this kind of scenario. There are other cryptographic notions collapsing in this model. Namely, the notion of invisibility in undeniable signatures (a device could convert a signature into some evidence that the

signature is true after running the verifier protocol), the notion of deniability of protocols in general (a device could prove that some protocol have been successfully executed), and the notion of receipt-freeness in electronic voting (a device could prove that someone casted a given ballot). On the other hand, some new types of protocols become feasible. We have already seen a *proof for having seen* a view. We can also make a *proof of ignorance*. A trusted agent can prove that a user ignores a secret key by showing that the device created the public key but never released the secret one. This could be used to prove that a user did not sign a document in a group signature, which would break the anonymity notion. Similarly, it could be used to prove that a user never decrypted some communication (namely that he did not open a sealed envelop), or did not cheat in some games.

We could still fix classical protocols by making honest participants themselves use trusted agents. One way to construct ONTAP protocols in the trusted agent model would consists of having the prover to use a trusted agent. In this situation, zero-knowledge becomes actually trivial: one can use a trusted agent receiving a (x,w) pair and checking that $(x,w) \in R$ holds to display x. The device becomes a zero-knowledge proof of knowledge of w. It is deniable in the sense that the user keeps the device and can deny having a device showing this statement.

However, people in the Caliph's realm certainly would not like that holding a trusted device would become a necessity for every day life. People would no longer be free to run private transactions. A consequence of Ali Baba's trial is that trusted devices have been forbidden to support the freedom of people.

This was Scheherazade's story adapted for an audience of people addicted to cell phones and popular music and video players. Hopefully, this would never happen in our civilized countries. (See Montesquieu revisited in [29])

Acknowledgements. The author would like to thank Jean-Jacques Quisquater to having told him (nearly) 1001 fascinating stories about cryptography in 1990. This paper is dedicated to him.

References

1. Machine Readable Travel Documents. PKI for Machine Readable Travel Documents offering ICC Read-Only Access. Version 1.1. International Civil Aviation Organization. 2004. http://www.icao.int/mrtd/download/technical.cfm
2. J. Baek, R. Safavi-Naini, W. Susilo. Universal Designated Verifier Signature Proof (or How to Efficiently Prove Knowledge of a Signature). In *Advances in Cryptology ASIACRYPT'05*, Chennai, India, Lecture Notes in Computer Science 3788, pp. 644–661, Springer-Verlag, 2005.
3. M. Bellare, A. Palacio. GQ and Schnorr Identification Schemes: Proofs of Security against Impersonation under Active and Concurrent Attacks. In *Advances in Cryptology CRYPTO'02*, Santa Barbara, California, U.S.A., Lecture Notes in Computer Science 2442, pp. 162–177, Springer-Verlag, 2002.
4. M. Bellare, T. Ristov. Hash Functions from Sigma Protocols and Improvements to VSH. In *Advances in Cryptology ASIACRYPT'08*, Melbourne, Australia, Lecture Notes in Computer Science 5350, pp. 125–142, Springer-Verlag, 2008.
5. G. Brassard, D. Chaum, C. Crépeau. Minimum Disclosure Proofs of Knowledge. *Journal of Computer and System Sciences*, vol. 37, pp. 156–189, 1988.
6. J. Camenisch, M. Michels. Confirmer Signature Schemes Secure against Adaptive Adversaries. In *Advances in Cryptology EUROCRYPT'00*, Brugge, Belgium, Lecture Notes in Computer Science 1807, pp. 243–258, Springer-Verlag, 2000.
7. D. Chaum. Designated Confirmer Signatures. In *Advances in Cryptology EUROCRYPT'94*, Perugia, Italy, Lecture Notes in Computer Science 950, pp. 86–91, Springer-Verlag, 1995.
8. D. Chaum, H. van Antwerpen. Undeniable Signatures. In *Advances in Cryptology CRYPTO'89*, Santa Barbara, California, U.S.A., Lecture Notes in Computer Science 435, pp. 212–217, Springer-Verlag, 1990.
9. R. Cramer, I. Damgård, P. MacKenzie. Efficient Zero-Knowledge Proofs of Knowledge Without Intractability Assumptions. In *Public Key Cryptography'00*, Melbourne, Australia, Lecture Notes in Computer Science 1751, pp. 354–373, Springer-Verlag, 2000.

10. Y. Desmedt. Subliminal-Free Authentication and Signature (Extended Abstract). In *Advances in Cryptology EUROCRYPT'88*, Davos, Switzerland, Lecture Notes in Computer Science 330, pp. 23–33, Springer-Verlag, 1988.
11. D. Dolev, C. Dwork, M. Naor. Nonmalleable Cryptography. *SIAM Reviews*, vol. 45(4), pp. 727–784, 2003.
12. A. Fiat, A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Advances in Cryptology CRYPTO'86*, Santa Barbara, California, U.S.A., Lecture Notes in Computer Science 263, pp. 186–194, Springer-Verlag, 1987.
13. O. Goldreich, S. Micali, A. Wigderson. Proofs that Yield Nothing but their Validity or all Languages in NP have Zero-Knowledge Proof Systems. *Communications of the ACM*, vol. 38, pp. 690–728, 1991.
14. L. C. Guillou, J.-J. Quisquater. A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Transmission and Memory. In *Advances in Cryptology EUROCRYPT'88*, Davos, Switzerland, Lecture Notes in Computer Science 330, pp. 123–128, Springer-Verlag, 1988.
15. L. C. Guillou, J.-J. Quisquater. A "Paradoxical" Identity-Based Signature Scheme Resulting from Zero-Knowledge. In *Advances in Cryptology CRYPTO'88*, Santa Barbara, California, U.S.A., Lecture Notes in Computer Science 403, pp. 216–231, Springer-Verlag, 1990.
16. M. Jakobsson, K. Sako, R. Impagliazzo. Designated Verifier Proofs and Their Applications. In *Advances in Cryptology EUROCRYPT'96*, Zaragoza, Spain, Lecture Notes in Computer Science 1070, pp. 143–154, Springer-Verlag, 1996.
17. J. Monnerat, S. Pasini, S. Vaudenay. Efficient Deniable Authentication for Signatures: Application to Machine-Readable Travel Document. In *Applied Cryptography and Network Security (ACNS'09)*, Paris-Rocquencourt, France, Lecture Notes in Computer Science 5536, pp. 272–291, Springer-Verlag, 2009.
18. P. Mateus, S. Vaudenay. On Privacy Losses in the Trusted Agent Model. Presented at the *EUROCRYPT'09* conference. Available on http://eprint.iacr.org/2009/286.pdf
19. P. Mateus, S. Vaudenay. On Tamper-Resistance from a Theoretical Viewpoint: The Power of Seals. In *Cryptographic Hardware and Embedded Systems CHES'09*, Lausanne, Switzerland, Lecture Notes in Computer Science 5747, pp. 411–428, Springer-Verlag, 2009.
20. J. Monnerat, S. Vaudenay, M. Vuagnoux. About Machine-Readable Travel Documents: Privacy Enhancement Using (Weakly) Non-Transferable Data Authentication. In *International Conference on RFID Security 2007*, Málaga, Spain, pp. 13–26, University of Málaga, 2008.
21. T. Okamoto, k. Ohta. How to Utilize the Randomness of Zero-Knowledge Proofs. In *Advances in Cryptology CRYPTO'90*, Santa Barbara, California, U.S.A., Lecture Notes in Computer Science 537, pp. 456–475, Springer-Verlag, 1991.
22. R. Pass. On Deniability in the Common Reference String and Random Oracle Model In *Advances in Cryptology CRYPTO'03*, Santa Barbara, California, U.S.A., Lecture Notes in Computer Science 2729, pp. 316–337, Springer-Verlag, 2003.
23. J.-J. Quisquater, M. Quisquater, M. Quisquater, M. Quisquater, L. Guillou, M. A. Guillou, G. Guillou, A. Guillou, G. Guillou, S. Guillou, T. A. Berson. How to Explain Zero-Knowledge Protocols to Your Children. In *Advances in Cryptology CRYPTO'89*, Santa Barbara, California, U.S.A., Lecture Notes in Computer Science 435, pp. 628–631, Springer-Verlag, 1990.
24. C.-P. Schnorr. Efficient Identification and Signatures for Smart Cards. In *Advances in Cryptology CRYPTO'89*, Santa Barbara, California, U.S.A., Lecture Notes in Computer Science 435, pp. 239–252, Springer-Verlag, 1990.
25. C.-P. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, vol. 4, pp. 161–174, 1991.
26. S. F. Shahandashti, R. Safavi-Naini, J. Baek. Concurrently-Secure Credential Ownership Proofs. In *ACM Symposium on Information, Computer and Communications Security (ASIACCS'07)*, Singapore, pp. 161–172, ACM Press, 2007.
27. R. Steinfeld, L. Bull, H. Wang, J. Pieprzyk. Universal Designated-Verifier Signatures. In *Advances in Cryptology ASIACRYPT'03*, Taipei, Taiwan, Lecture Notes in Computer Science 2894, pp. 523–542, Springer-Verlag, 2003.
28. S. Vaudenay. E-Passport Threats. *IEEE Security & Privacy*, vol. 5(6), pp. 61–64, 2007.
29. S. Vaudenay. *La Fracture Cryptographique*, Focus Science, Presses Polytechniques et Universitaires Romandes, 2010.
30. S. Vaudenay, M. Vuagnoux. About Machine-Readable Travel Documents. *Journal of Physics: Conference Series*, vol. 77, num. 012006, 2007. http://www.iop.org/EJ/article/1742-6596/77/1/012006/jpconf7i_77_012006.pdf