Cryptographic reverse firewalls for interactive proof systems

DOI:
10.1016/j.tcs.2020.11.043

Cryptographic Reverse Firewalls for Interactive Proof Systems

Chaya Ganesh
Department of Computer Science and Automation, Indian Institute of Science, India
chaya@iisc.ac.in

Bernardo Magri
Department of Computer Science, Aarhus University, Denmark
magri@cs.au.dk

Daniele Venturi
Department of Computer Science, Sapienza University of Rome, Italy
venturi@di.uniroma1.it

Abstract

We study interactive proof systems (IPSes) in a strong adversarial setting where the machines of *honest parties* might be corrupted and under control of the adversary. Our aim is to answer the following, seemingly paradoxical, questions:

- Can Peggy convince Vic of the veracity of an NP statement, without leaking any information about the witness even in case Vic is malicious and Peggy does not trust her computer?
- Can we avoid that Peggy fools Vic into accepting false statements, even if Peggy is malicious and Vic does not trust her computer?

At EUROCRYPT 2015, Mironov and Stephens-Davidowitz introduced cryptographic reverse firewalls (RFs) as an attractive approach to tackling such questions. Intuitively, a RF for Peggy/Vic is an external party that sits between Peggy/Vic and the outside world and whose scope is to sanitize Peggy's/Vic's incoming and outgoing messages in the face of subversion of her/his computer, *e.g.* in order to destroy subliminal channels.

In this paper, we put forward several natural security properties for RFs in the concrete setting of IPSes. As our main contribution, we construct efficient RFs for different IPSes derived from a large class of Sigma protocols that we call *malleable*.

A nice feature of our design is that it is completely transparent, in the sense that our RFs can be directly applied to already deployed IPSes, without the need to re-implement them.

2012 ACM Subject Classification Security and privacy → Cryptography

Keywords and phrases Subversion, Algorithm substitution attacks, Cryptographic reverse firewalls, Interactive proofs, Zero knowledge

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.55

Related Version A full version of the paper is available at https://eprint.iacr.org/2020/204.

Funding *Bernardo Magri*: This work was supported by Concordium Blockchain Research Center, Aarhus University, Denmark

1 Introduction

An interactive proof system (IPS) $\Pi = (\mathsf{P}, \mathsf{V})$ allows a prover P to convince a verifier V about the veracity of a public statement $x \in \mathcal{L}$, where $\mathcal{L}$ is an NP language and where both P and V are modeled as interactive PPT machines. The prover is facilitated by possessing a witness w to the fact that, indeed, $x \in \mathcal{L}$, and the interaction with the verifier may consist of several rounds of communication, at the end of which the verifier outputs a verdict on the membership of x in $\mathcal{L}$.

In order to be useful, an IPS should satisfy the following properties:

- *Completeness:* If $x \in \mathcal{L}$, the honest prover (almost) always convinces the honest verifier.
- *Soundness:* If $x \notin \mathcal{L}$, no (computationally bounded) malicious prover can convince the honest verifier that $x \in \mathcal{L}$. An even stronger guarantee, known as *knowledge soundness* [9], is to require that the only way a prover can convince the honest verifier that $x \in \mathcal{L}$ is to "know" a valid witness w corresponding to x. Such proofs[1] are called *proofs of knowledge* (PoKs).
- *Zero Knowledge (ZK):* A valid proof reveals nothing beyond the fact that $x \in \mathcal{L}$, and thus in particular it leaks no information about the witness w, even in case the proof is conducted in the presence of a (computationally bounded) malicious verifier [36]. A weaker guarantee, known as *witness indistinguishability* (WI) [24], is that, whenever there are multiple witnesses attesting that $x \in \mathcal{L}$, no (computationally bounded) malicious verifier can distinguish whether a proof is conducted using either of two witnesses.

One of the motivations for studying IPSes with the above properties is that they are ubiquitous in cryptography, with applications ranging from identification protocols [24], blind digital signatures [42], and electronic voting [16], to general-purpose maliciously secure multi-party computation [35].

1.1 Sigma Protocols

While WI/ZK PoKs exist for all of NP, based on minimal cryptographic assumptions [23, 34, 33], efficiency is a different story. Fortunately, it is possible to design practical interactive proofs for specific languages, typically in the form of so-called Sigma protocols. Briefly, a Sigma protocol is a special type of IPS consisting of just three rounds, where the prover sends a first message α (the commitment), the verifier sends a random string β (the challenge), and finally the prover forwards a last message γ (the response). Sigma protocols satisfy two main properties: The first one, known as *special soundness*, is a strong form of knowledge soundness; the second one, known as *honest-verifier zero knowledge* (HVZK), is a weak form of the zero knowledge property that only holds against honest-but-curious verifiers.

The applications of Sigma protocols to cryptographic constructions are countless (see, *e.g.*, [25, 17, 48, 22, 43]). These results are perhaps surprising, as Sigma protocols only satisfy HVZK and thus guarantee no security in the presence of malicious verifiers. In some cases, the solution to this apparent paradox is due to a beautiful technique put forward by Cramer, Damgård, and Schoenmakers [15], which allows to add WI to any Sigma protocol. Moreover, it is relatively easy to transform any Sigma protocol into an interactive ZK PoK at the cost of adding a single round of interaction [33].

[1] Sometimes, the term "proof" is used to refer to statistically sound IPSes, while computationally sound IPSes are typically called "arguments".

1.2 Our Question

The standard definitions of security for IPSes (implicitly) rely on the assumption that honest parties can fully trust their machines. In practice, however, such an assumption may just be too optimistic, as witnessed by the revelations of Edward Snowden about subversion of cryptographic standards [45, 7], and in light of the numerous (seemingly accidental) bugs in widespread pieces of cryptographic software [38, 1, 2].

Motivated by the above incidents, we ask the following question which constitutes the main source of inspiration for this work:

> *Can we design practical interactive proofs that remain secure even if the machines of the honest parties running them have been tampered with?*

In order to see why the above question is well motivated and not trivial, let us analyze the dramatic consequences of subverting the prover of ZK IPSes. Clearly, the problem of subversion-resistant interactive zero knowledge is just impossible in its utmost generality, as a subverted prover could just reveal the witness to the verifier. However, one may argue that these kind of attacks are easily detectable, and thus can be avoided.

The problem becomes more interesting if we restrict the subversion to be *undetectable*, as suggested by Bellare, Paterson, and Rogaway [11] in their seminal work on subversion of symmetric encryption, where the authors show how to subvert any sufficiently randomized cipher in an undetectable manner, using rejection sampling. A moment of reflection shows that their attack can be adapted to the case of IPSes.[2] The solution proposed by [11] is to rely on deterministic symmetric encryption. Unfortunately, this approach is not viable for the case of IPSes, as it is well-known that interactive proofs with deterministic provers can be zero knowledge only for trivial languages [32, §4.5].

Reverse firewalls

The above described undetectable attacks show that the problem of designing IPSes that remain secure even when run on untrusted machines is simply impossible if we are not willing to make any further assumption. In this paper, we study how to tackle subversion attacks against interactive proofs in the framework of "cryptographic reverse firewalls (RFs)", introduced by Mironov and Stephens-Davidowitz [40]. In such a setting, both the prover and the verifier are equipped with their own RF W, also modeled as an interactive PPT machine, whose scope is solely to sanitize the parties' incoming and outgoing messages in the face of subversion.

Importantly, neither the prover nor the verifier put any trust in the RF, meaning that they are not allowed to share secrets with the firewall itself. The hope is that an uncorrupted[3] RF can provide meaningful security guarantees even in case the honest prover's and/or verifier's machines have been tampered with. Note that a RF can never "create security", as it does not even know the inputs to the protocol, but at best can preserve the security guarantees satisfied by the initial IPS. At the same time, the RF should not ruin the functionality of the

[2] In particular, a subverted prover with a hardwired secret key k for a pseudorandom function $F_k(\cdot)$, could sample the random coins $r^{(i)}$ needed to generate the honest prover's message $m^{(i)}$ (for round $i \in \mathbb{N}$) multiple times, until $F_k(m^{(i)})$ leaks one bit of the witness. This attack works provided that at least one of the prover's messages has high-enough min-entropy.

[3] Clearly, if both the machine of the honest party and the firewall are corrupted, there is no hope for security. On the other hand, in case the machine is honest and the firewall is corrupt, the underlying protocol is still secure, since we can simply think of the RF as being part of the adversary [21].

underlying IPS, in the sense that the sanitized IPS should still work in case no subversion takes place.

Mironov and Stephens-Davidowitz construct general-purpose RFs that can be used in order to preserve both functionality and security of any two-party protocol. It is important to note that since ZK/WI IPSes are a special case of secure two-party computation, their RF constructions already seem to solve our problem.[4] However, the solutions in [40] are not practical. In particular, one of their RFs increases the round complexity of the initial IPS, and, more importantly, it requires to carry out the underlying IPS in the encrypted domain, thus requiring to completely change the original protocol. In contrast, we seek constructions of RFs that can be applied directly to existing IPSes, without adding any overhead, and without the need to re-implement them.

2 Reverse Firewalls for Interactive Proofs

In this section, we give security definitions for RFs applied to IPSes. Our notions can be seen as special cases of the generic framework by Mironov and Stephens-Davidowitz [40], who defined security of RFs for the more general case of arbitrary two-party protocols.

Let $\Pi = (\mathsf{P}, \mathsf{V})$ be an IPS for a relation $\mathcal{R}$. A cryptographic reverse firewall is an external party W that can be attached either to the prover P or to the verifier V, whose scope is to sanitize incoming and outgoing messages in the face of parties' subversion. Importantly, the RF is allowed to keep its own state but cannot share state with any of the parties. Similarly to [40], we model an interactive Turing machine M as a triple of algorithms $\mathsf{M} := (\mathsf{M}_{\mathsf{nxt}}, \mathsf{M}_{\mathsf{rec}}, \mathsf{M}_{\mathsf{out}})$ specified as follows: (i) Algorithm $\mathsf{M}_{\mathsf{nxt}}$ takes as input the current state and outputs the next message to be sent; (ii) Algorithm $\mathsf{M}_{\mathsf{rec}}$ takes as input an incoming message, and updates the state; (iii) Algorithm $\mathsf{M}_{\mathsf{out}}$ takes as input the final state at the completion of the protocol, and returns a bit.

▶ **Definition 1** (RF for IPSes). *Let $\Pi = (\mathsf{P}, \mathsf{V})$ be an IPS for a relation $\mathcal{R}$. A cryptographic reverse firewall (RF) for Π is a stateful algorithm W that takes as input a message, its state, and outputs a sanitized message, together with an updated state. For an interactive Turing machine $\mathsf{M} = (\mathsf{M}_{\mathsf{nxt}}, \mathsf{M}_{\mathsf{rec}}, \mathsf{M}_{\mathsf{out}}) \in \{\mathsf{P}, \mathsf{V}\}$, and RF W, the sanitized machine $\mathsf{W} \circ \mathsf{M} := \widehat{\mathsf{M}} = (\widehat{\mathsf{M}}_{\mathsf{nxt}}, \widehat{\mathsf{M}}_{\mathsf{rec}}, \widehat{\mathsf{M}}_{\mathsf{out}})$ is specified as follows:*

$$\begin{aligned}
\widehat{\mathsf{M}}_{\mathsf{nxt}}(\sigma) &:= \mathsf{W}(\mathsf{M}_{\mathsf{nxt}}(\sigma)) \\
\widehat{\mathsf{M}}_{\mathsf{rec}}(\sigma, m) &:= \mathsf{M}_{\mathsf{rec}}(\sigma, \mathsf{W}(m)) \\
\widehat{\mathsf{M}}_{\mathsf{out}}(\sigma) &:= \mathsf{M}_{\mathsf{out}}(\sigma).
\end{aligned}$$

As our first contribution, we put forward several natural properties that a RF for an IPS might satisfy. In particular, we consider the following notions (see the full version [29] for more formal definitions).

- *Completeness preservation:* The sanitized IPS (*i.e.*, the IPS obtained by sanitizing both the honest prover's and the honest verifier's messages) still satisfies completeness.
- *Strong soundness preservation:* Whenever $x \notin \mathcal{L}$, no malicious prover can convince the verifier that $x \in \mathcal{L}$, even if the verifier's implementation has been arbitrarily subverted.

[4] At least to some extent, since, strictly speaking, their results for IPSes are incomparable to ours. We refer the reader to §5.1 for more details.

- *Strong ZK preservation:* A valid proof reveals nothing beyond the fact that $x \in \mathcal{L}$, even in case the proof is conducted in the presence of a malicious verifier talking to a prover whose implementation has been arbitrarily subverted.
- *Strong WI preservation:* Whenever there are multiple witnesses attesting that $x \in \mathcal{L}$, no malicious verifier talking to a prover whose implementation has been arbitrarily subverted can distinguish whether a proof is conducted using either of two witnesses.
- *Strong exfiltration resistance for the prover (resp. verifier):* Transcripts produced by running the sanitized IPS in the presence of a malicious verifier (resp. prover) talking to a prover (resp. verifier) whose implementation has been arbitrarily subverted are indistinguishable to transcripts produced by running the sanitized IPS in the presence of a malicious verifier (resp. prover) talking to the honest prover (resp. verifier).

For each of the above properties (except for completeness), we also consider a weak variant which only holds w.r.t. *functionality-maintaining* provers/verifiers. Intuitively, a prover is functionality maintaining if, upon input a valid statement/witness pair, it still convinces the honest verifier with overwhelming probability. Similarly, a verifier is functionality maintaining if, upon input a valid statement, it still accepts with overwhelming probability in a protocol run with the honest prover.

What is possible and what is impossible

A moment of reflection shows that soundness preservation is impossible to achieve. In fact, an arbitrarily subverted verifier might always[5] output one, thus automatically accepting both true and false statements. Such a verifier is still functionality maintaining,[6] and thus this simple attack even rules out *weak* soundness preservation. One way to circumvent this impossibility would be to only consider *partial subversion*, *i.e.* split the verifier into two components, one for computing the next messages in the protocol, and the other one for determining the final verdict on the veracity of a statement; hence, assume the latter component to be untamperable.

Turning to subversion of the prover, consider the subverted prover that always outputs the all-zero string. The soundness property of the underlying IPS implies that, for any RF and for any *false* statement $x \notin \mathcal{L}$, a sanitized transcript in this case can never be accepting. Moreover, assuming the language $\mathcal{L}$ is non-trivial, the latter holds true even in case x is a *true* statement, which in turn rules out strong exfiltration resistance. For similar reasons, strong ZK/WI preservation are also impossible to achieve.

3 Firewall Constructions from Malleable Sigma Protocols

As our second contribution, we formalize a class of Sigma protocols which admit simple, and very efficient, RFs for the prover. (See the full version [29] for similar constructions dealing with functionality-maintaining subversion of the verifier.) The main idea is to use the RF to re-randomize the prover's messages, in order to destroy any potential subliminal channel signaling information about the witness. The difficulty, though, is that such re-randomization must be carried out without knowing a witness, and while at the same time preserving the completeness property of the underlying IPS.

---

[5] If one insists on undetectability, the subverted verifier may output 1 upon some hard-wired, randomly chosen, false statement $\overline{x} \notin \mathcal{L}$.

[6] The latter is because completeness is a guarantee that only concerns true statements.

Prover(x, w)		**Reverse Firewall**		**Verifier**(x)
$\alpha = \mathsf{P}_1(x, w; a)$				
	$\xrightarrow{\alpha}$	$(\widehat{\alpha}, \sigma) \leftarrow\!\!\$\ \mathsf{Maul}(\alpha)$	$\xrightarrow{\widehat{\alpha}}$	
				$\beta \leftarrow\!\!\$\ \{0,1\}^\ell$
	$\xleftarrow{\beta}$		$\xleftarrow{\beta}$	
$\gamma = \mathsf{P}_2(x, w, \beta, a)$				
	$\xrightarrow{\gamma}$	$\widehat{\gamma} = \mathsf{Bal}(\gamma, \sigma)$	$\xrightarrow{\widehat{\gamma}}$	
				$\mathsf{V}(x, (\widehat{\alpha}, \beta, \widehat{\gamma})) \stackrel{?}{=} 1$

Figure 1 Cryptographic reverse firewall for a malleable Sigma protocol

For the sake of concreteness, let us describe our firewall applied to the classical Sigma protocol for proving knowledge of a discrete logarithm [49]. Here, the statement consists of a description of a cyclic group $\mathbb{G}$ with generator g and prime order q, together with a value $x \in \mathbb{G}$ such that $x = g^w$ for some $w \in \mathbb{Z}_q$. The prover's first message is a random group element $\alpha = g^a \in \mathbb{G}$. Finally, the prover's last message is $\gamma = a - w \cdot \beta$, where $\beta \in \mathbb{Z}_q$ is the verifier's challenge; the verifier accepts (α, β, γ) if and only if $g^\gamma = \alpha \cdot x^{-\beta}$. Our RF sanitizes the messages α and γ from a possibly subverted implementation of the prover as follows:

$$\widehat{\alpha} = \alpha \cdot g^\sigma$$
$$\widehat{\gamma} = \gamma + \sigma,$$

for random $\sigma \in \mathbb{Z}_q$. Note that $g^{\widehat{\gamma}} = g^a \cdot g^\sigma \cdot x^{-\beta} = \widehat{\alpha} \cdot x^{-\beta}$, and thus the RF preserves completeness.

We now sketch the proof of weak HVZK preservation. Observe that for any $\widetilde{\alpha} = g^{\widetilde{a}}$ sent by a functionality-maintaining subverted prover, the distribution of $\widehat{\alpha} = g^{\widetilde{a}+\sigma}$ is uniform over $\mathbb{G}$ and independent of $\widetilde{\alpha}, \widetilde{a}$, and in fact it is identical to the distribution of α in an honest run of the original Sigma protocol (without the firewall). As for $\widehat{\gamma}$, note that if there would be two possible values γ, γ' which make both $\tau = (\alpha, \beta, \gamma)$ and $\tau' = (\alpha, \beta, \gamma')$ valid transcripts, the choice of which response to pick could be used by a functionality-maintaining subverted prover as a subliminal channel signaling information about the witness. Hence, we exploit the fact that for any prefix α, β, there exists a unique response γ such that the verifier accepts upon input x and (α, β, γ).

It follows that the distribution of $\widehat{\gamma}$ is identical to that of γ in an honest run of the original Sigma protocol (without the firewall). Putting it all together, we have shown that the distribution of a sanitized transcript $\widehat{\tau} = (\widehat{\alpha}, \beta, \widehat{\gamma})$ is identical to the distribution of an honest transcript $\tau = (\alpha, \beta, \gamma)$. Thus, weak HVZK preservation follows by the fact that Schnorr's Sigma protocol is HVZK.

3.1 HVZK Preservation

Let us now explain how to generalize the above idea to a large class of Sigma protocols that we call *malleable*. In what follows, given a Sigma protocol $\Sigma = (\mathsf{P}, \mathsf{V})$, we denote by P_1 and P_2 the algorithms that compute, respectively, the first prover's message α, and the last prover's message (response) γ. The challenge space is represented[7] as $\{0,1\}^\ell$, so that there are 2^ℓ possible challenges, and we write V for the algorithm that the verifier runs upon statement x and transcript τ to make its final decision. Let $\mathcal{A}$ be the space of all possible prover's first messages; we assume that membership in $\mathcal{A}$ can be tested efficiently, so that V always outputs $\perp$ whenever $\alpha \notin \mathcal{A}$.

As for the case of Schnorr's Sigma protocol, an additional requirement that we need is that the prover's responses are unique, meaning that for all $x \in \mathcal{L}$, and for any $\alpha \in \mathcal{A}$ and $\beta \in \{0,1\}^\ell$, there exists at most one[8] value γ such that $\mathsf{V}(x, (\alpha, \beta, \gamma)) = 1$.

Intuitively, a Sigma protocol is malleable if there exists an efficient algorithm Maul for randomizing the prover's first message α into a value $\widehat{\alpha}$ which is distributed identically to the first message of an honest prover. Moreover, for any challenge β, given the coins used to randomize α and any response γ yielding a valid transcript $\tau = (\alpha, \beta, \gamma)$, there exists an efficient algorithm Bal for computing a balanced response $\widehat{\gamma}$ such that $(\widehat{\alpha}, \beta, \widehat{\gamma})$ is also valid. As we show in the full version [29], many natural Sigma protocols are already malleable. In particular, the latter holds true for Maurer's unifying protocol [39], which includes the protocols by Fiat-Shamir [25], Guillou-Quisquater [37], Schnorr [49], Okamoto [41], and many others as special cases.

Our RF construction is depicted in Fig. 1. Intuitively, the firewall uses the malleability property of the underlying Sigma protocol in order to re-randomize the prover's first and last messages, in such a way that a functionality-maintaining subverted prover cannot signal information about the witness through them. The theorem below, whose proof appears in the full version [29], establishes its security.

▶ **Theorem 2.** *Let $\Sigma = (\mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2), \mathsf{V})$ be a malleable Sigma protocol with unique responses, for a relation $\mathcal{R}$. The RF W of Fig. 1 preserves completeness, and is weakly HVZK preserving for the prover.*

3.2 ZK Preservation

As Sigma protocols are not in general zero knowledge, there is no hope to prove that the above firewall weakly preserves ZK. However, a standard trick [33] allows to transform any Sigma protocol into a 5-round IPS satisfying ZK. The idea is to let the prover send the public key pk of a commitment scheme $(\mathsf{Gen}, \mathsf{Com}, \mathsf{Open})$ during the first round. Then, during the second round, the verifier forwards to the prover a commitment c to the challenge β. Finally, the Sigma protocol is executed as before with the difference that the verifier also needs to open the commitment, with the prover aborting if the opening is invalid. We depict such a modified protocol in Fig. 2.

In order to build a RF for this IPS, we need to sanitize the additional messages from the (possibly subverted, but functionality-maintaining) prover.[9] We do so by relying on a special type of *key-malleable* commitment, which intuitively allows to maul any public key pk (via an algorithm $\mathsf{MaulKey}$) into a uniformly random public key $\widehat{pk}$, in such a way that, given a commitment c with opening d w.r.t. $\widehat{pk}$, it is possible to map (c, d) into a commitment $\widehat{c}$ with opening $\widehat{d}$ w.r.t. pk, without changing the message inside the commitment. We

[7] In the case of Schnorr's Sigma protocol, the challenge space is a cyclic group. However, we can embed such group in $\{0,1\}^\ell$ for some $\ell \in \mathbb{N}$.

[8] This property is met by many natural Sigma protocols, and was already considered in several previous works [26, 22, 51].

[9] The other messages are sanitized as before, *i.e.* we still exploit the fact that the underlying Sigma protocol is malleable.

Prover(x, w)		**Verifier**(x)
$pk \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$	$\xrightarrow{pk}$	
		$\beta \leftarrow\!\!\$\ \{0,1\}^\ell$ $(c, d) \leftarrow\!\!\$\ \mathsf{Com}(pk, \beta)$
	$\xleftarrow{c}$	
$\alpha = \mathsf{P}_1(x, w; a)$		
	$\xrightarrow{\alpha}$	
	$\xleftarrow{d}$	
$\beta = \mathsf{Open}(pk, c, d)$ If $\beta \neq \perp$, then $\gamma = \mathsf{P}_2(x, w, \alpha, \beta, a)$ Else, $\gamma = \perp$		
	$\xrightarrow{\gamma}$	
		$\mathsf{V}(x, (\alpha, \beta, \gamma)) \stackrel{?}{=} 1$

Figure 2 Sigma protocol compiled with standard techniques to obtain full zero knowledge

denote by MaulCom and BalOpen, respectively, the algorithms for mauling the commitment c and the opening d, and additionally require that the distribution of mauled public keys and commitments is identical, respectively, to that of honestly computed public keys and commitments. As we show in the full version [29], the standard Pedersen's commitment [44] is easily seen to be key malleable, thus yielding a concrete instantiation under the Discrete Logarithm assumption.

Our RF for the protocol of Fig. 2 is depicted in Fig. 3. The theorem below, whose proof appears in the full version [29], establishes its security.

▶ **Theorem 3.** *Let $\Sigma = (\mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2), \mathsf{V})$ be a malleable Sigma protocol with unique responses, for a relation $\mathcal{R}$. Let $\Gamma = (\mathsf{Gen}, \mathsf{Com}, \mathsf{Open})$ be a key-malleable commitment scheme with message space $\{0,1\}^\ell$. The RF W of Fig. 3 preserves completeness, and moreover is weakly exfiltration resistant and weakly zero-knowledge preserving for the prover.*

▶ Remark 4 (On knowledge soundness). The IPS of Fig. 2 satisfies soundness, but is not in general a proof of knowledge. However, we would like to note that the prover's firewall still works for the standard transformation of a Sigma protocol into a zero-knowledge proof of knowledge. In such a transformation, a *trapdoor* commitment scheme is used to commit to the verifier's challenge. Then, after the verifier decommits, the prover sends the trapdoor to the verifier. This allows an extractor to learn the trapdoor, rewind the prover, and open the commitment to a different challenge, thus learning the response for two different challenges, which allows it to obtain a witness using special soundness.

The prover's RF for this protocol stays the same, except that it additionally needs to provide a trapdoor for the mauled public key $\widehat{pk}$ given a trapdoor for the original public key pk. This is possible, for instance, using Pedersen's commitment, where given a public key $pk = (g, h = g^k)$ with trapdoor k, we can maul the key to $(\widehat{g} = g^{t_1}, \widehat{h} = h^{t_2})$ for random t_1, t_2. Given the trapdoor k for the key pk, the trapdoor for the mauled key $\widehat{pk}$ can be computed as $t_2 t_1^{-1} k$.

Prover(x,w)		**ZK Reverse Firewall**		**Verifier**(x)
$pk \leftarrow_\$ \mathsf{Gen}(1^\lambda)$	$\xrightarrow{pk}$	$(\widehat{pk},\rho) \leftarrow_\$ \mathsf{MaulKey}(pk)$	$\xrightarrow{\widehat{pk}}$	
				$\beta \leftarrow_\$ \{0,1\}^\ell$ $(c,d) \leftarrow_\$ \mathsf{Com}(\widehat{pk},\beta)$
	$\xleftarrow{\widehat{c}}$	$\widehat{c} \leftarrow_\$ \mathsf{MaulCom}(pk,c,\rho)$	$\xleftarrow{c}$	
$\alpha = \mathsf{P}_1(x,w;a)$				
	$\xrightarrow{\alpha}$	$(\widehat{\alpha},\sigma) \leftarrow_\$ \mathsf{Maul}(\alpha)$	$\xrightarrow{\widehat{\alpha}}$	
	$\xleftarrow{\widehat{d}}$	$\widehat{d} = \mathsf{BalOpen}(d,\rho)$	$\xleftarrow{d}$	
$\beta = \mathsf{Open}(pk,\widehat{c},\widehat{d})$ If $\beta \neq \perp$, then $\gamma = \mathsf{P}_2(x,w,\beta,a)$				
	$\xrightarrow{\gamma}$			
		$\beta = \mathsf{Open}(pk,\widehat{c},\widehat{d})$ If $\beta = \perp$, then $\widehat{\gamma} = \perp$ Else, $\widehat{\gamma} = \mathsf{Bal}(\gamma,\sigma)$		
			$\xrightarrow{\widehat{\gamma}}$	
				$\mathsf{V}(x,(\widehat{\alpha},\beta,\widehat{\gamma})) \stackrel{?}{=} 1$

Figure 3 Prover's RF for the protocol in Fig. 2

4 Firewalls for Proving Compound Statements

In this section, we show how to construct firewalls for Sigma protocols that prove compound statements.

Given two Sigma protocols Σ_0 and Σ_1 for NP languages $\mathcal{L}_0$ and $\mathcal{L}_1$, it is easy to obtain a Sigma protocol Σ_{AND} for the NP language $\mathcal{L}_{\mathsf{AND}} = \{(x_0,x_1) : x_0 \in \mathcal{L}_0 \wedge x_1 \in \mathcal{L}_1\}$ by simply running Σ_0 and Σ_1 in parallel, with the verifier sending a single challenge. In a similar vein, the OR technique by Cramer, Damgård, and Schoenmakers [15] allows to obtain a Sigma protocol Σ_{OR} for the NP language $\mathcal{L}_{\mathsf{OR}} = \{(x_0,x_1) : x_0 \in \mathcal{L}_0 \vee x_1 \in \mathcal{L}_1\}$. Importantly, if Σ_0 and Σ_1 are both perfect HVZK, Σ_{OR} satisfies perfect WI. On the other hand, Garay *et al.* [30] showed that if Σ_0 and Σ_1 are computational HVZK, Σ_{OR} satisfies computational WI, although the latter holds only in case both statements x_0, x_1 in the definition of language $\mathcal{L}_{\mathsf{OR}}$ are true (but the prover knows either a witness for x_0 or for x_1).

As long as Σ_0 and Σ_1 are malleable, it is easy to build RFs for Σ_{AND} and Σ_{OR} using our techniques. The RF for Σ_{AND} weakly preserves HVZK, whereas the RF for Σ_{OR} weakly preserves both HVZK and WI.

4.1 AND Composition

Given x_0, x_1, a prover wishes to prove to a verifier that $x_0 \in \mathcal{L}_0$ and $x_1 \in \mathcal{L}_1$. More precisely, consider the derived relation:

$$\mathcal{R}_{\mathsf{AND}} = \{((x_0,x_1),(w_0,w_1)) : (x_0,w_0) \in \mathcal{R}_0 \wedge (x_1,w_1) \in \mathcal{R}_1\}.$$

Let $\Sigma_0 = ((\mathsf{P}_1^0,\mathsf{P}_2^0),\mathsf{V}^0)$ (resp. $\Sigma_1 = ((\mathsf{P}_1^1,\mathsf{P}_2^1),\mathsf{V}^1)$) be a Sigma protocol for language $\mathcal{L}_0$ (resp. $\mathcal{L}_1$). A Sigma protocol Σ_{AND} for the relation $\mathcal{R}_{\mathsf{AND}}$ can be obtained by running the two provers of Σ_0 and Σ_1 in parallel, with the verifier sending a single challenge for both executions. Fig. 4 shows a RF for the prover of Σ_{AND}, assuming that both Σ_0 and Σ_1 are malleable. We prove the following result, whose proof appears in the full version [29].

Prover$((x_0,x_1),(w_0,w_1))$		**Reverse Firewall**		**Verifier**(x_0,x_1)
$\alpha_0 = \mathsf{P}_1^0(x_0,w_0;a_0)$ $\alpha_1 = \mathsf{P}_1^1(x_1,w_1;a_1)$				
	$\xrightarrow{\alpha_0,\alpha_1}$			
		$(\widehat{\alpha}_0,\sigma_0) \leftarrow\$ \mathsf{Maul}_0(\alpha_0)$ $(\widehat{\alpha}_1,\sigma_1) \leftarrow\$ \mathsf{Maul}_1(\alpha_1)$		
			$\xrightarrow{\widehat{\alpha}_0,\widehat{\alpha}_1}$	
				$\beta \leftarrow \{0,1\}^\ell$
	$\xleftarrow{\beta}$		$\xleftarrow{\beta}$	
$\gamma_0 = \mathsf{P}_2^0(x_0,w_0,\beta,a_0)$ $\gamma_1 = \mathsf{P}_2^1(x_1,w_1,\beta,a_1)$				
	$\xrightarrow{\gamma_0,\gamma_1}$			
		$\widehat{\gamma}_0 = \mathsf{Bal}_0(\gamma_0,\sigma_0)$ $\widehat{\gamma}_1 = \mathsf{Bal}_1(\gamma_1,\sigma_1)$		
			$\xrightarrow{\widehat{\gamma}_0,\widehat{\gamma}_1}$	
				$\mathsf{V}^0(x_0,(\widehat{\alpha}_0,\beta,\widehat{\gamma}_0)) \stackrel{?}{=} 1$ $\mathsf{V}^1(x_1,(\widehat{\alpha}_1,\beta,\widehat{\gamma}_1)) \stackrel{?}{=} 1$

Figure 4 Reverse firewall for the AND composition of Sigma protocols

▶ **Theorem 5.** *Let $\Sigma_0 = (\mathsf{P}^0 = (\mathsf{P}_1^0,\mathsf{P}_2^0),\mathsf{V}^0)$ and $\Sigma_1 = (\mathsf{P}^1 = (\mathsf{P}_1^1,\mathsf{P}_2^1),\mathsf{V}^1)$ be malleable Sigma protocols with unique responses, for relations $\mathcal{R}_0$ and $\mathcal{R}_1$. The RF W of Fig. 4 preserves completeness, and is weakly HVZK preserving for the prover of the Sigma protocol Σ_{AND} for relation $\mathcal{R}_{\mathsf{AND}}$.*

4.2 OR Composition

Given x_0, x_1, a prover wishes to prove to a verifier that either $x_0 \in \mathcal{L}_0$ or $x_1 \in \mathcal{L}_1$ (without revealing which one is the case). More precisely, consider the derived relation

$$\mathcal{R}_{\mathsf{OR}} = \{((x_0,x_1),w) : \ (x_0,w) \in \mathcal{R}_0 \vee (x_1,w) \in \mathcal{R}_1\}.$$

Let $\Sigma_0 = ((\mathsf{P}_1^0,\mathsf{P}_2^0),\mathsf{V}^0)$ (resp. $\Sigma_1 = ((\mathsf{P}_1^1,\mathsf{P}_2^1),\mathsf{V}^1)$) be a Sigma protocol for language $\mathcal{L}_0$ (resp. $\mathcal{L}_1$); we denote by S^0 (resp. S^1) the HVZK simulator for Σ_0 (resp. Σ_1). A Sigma protocol Σ_{OR} for the relation $\mathcal{R}_{\mathsf{OR}}$ has been constructed for the first time in [15], where the authors showed that Σ_{OR} satisfies both (perfect) special HVZK and (perfect) WI. We describe the protocol Σ_{OR} in Fig. 5, and depict our RF for the prover in Fig. 6.

As in the case of AND composition, we still rely on the fact that the input Sigma protocols Σ_0, Σ_1 are malleable. An additional difficulty, however, stems from the fact that a functionality maintaining prover could now try to change the distribution of the challenges β_0, β_1 in such a way that, even if $\beta_0 \oplus \beta_1 = \beta$, the pair (β_0, β_1) signals some information about the witness w or about the hidden bit b. Intuitively, the RF in Fig. 6 tackles this attack by randomizing the challenges β, β_0, β_1. The latter requires a different form of malleability from the underlying Sigma protocols, which we dub *instance-dependent malleability*, where it should be possible to maul the prover's first message in such a way that we can later balance the prover's last message as well as the verifier's challenge.

For the RF in Fig. 6, we prove the following result, whose proof appears in the full version [29] of this paper.

$\mathbf{Prover}((x_0, x_1), w)$		$\mathbf{Verifier}(x_0, x_1)$
$\alpha_b = \mathsf{P}_1^b(x_b, w; a)$ $(\alpha_{1-b}, \beta_{1-b}, \gamma_{1-b}) \leftarrow\!\!\$\ \mathsf{S}^{1-b}(x_{1-b})$		
	$\xrightarrow{\alpha_0, \alpha_1}$	
		$\beta \leftarrow\!\!\$\ \{0,1\}^\ell$
	$\xleftarrow{\beta}$	
$\beta_b = \beta \oplus \beta_{1-b}$ $\gamma_b = \mathsf{P}_2^b(x_b, w, \beta_b, a)$		
	$\xrightarrow{\beta_0, \beta_1, \gamma_0, \gamma_1}$	
		$\beta \stackrel{?}{=} \beta_0 \oplus \beta_1$ $\mathsf{V}^0(x_0, (\alpha_0, \beta_0, \gamma_0)) \stackrel{?}{=} 1$ $\mathsf{V}^1(x_1, (\alpha_1, \beta_1, \gamma_1)) \stackrel{?}{=} 1$

Figure 5 OR composition of Sigma protocols, where $b \in \{0,1\}$ is s.t. $(x_b, w) \in \mathcal{R}_b$.

$\mathbf{Prover}((x_0, x_1), w)$		$\mathbf{Reverse\ Firewall}(x_0, x_1)$		$\mathbf{Verifier}(x_0, x_1)$
$\alpha_b = \mathsf{P}_1^{1-b}(x_b, w; a)$ $(\alpha_{1-b}, \beta_{1-b}, \gamma_{1-b}) \leftarrow\!\!\$\ \mathsf{S}^{1-b}(x_{1-b})$				
	$\xrightarrow{\alpha_0, \alpha_1}$			
		$\rho_0, \rho_1 \leftarrow\!\!\$\ \{0,1\}^\ell$ $(\widehat{\alpha}_0, \sigma_0) \leftarrow\!\!\$\ \mathsf{Maul}_0(x_0, \alpha_0, \rho_0)$ $(\widehat{\alpha}_1, \sigma_1) \leftarrow\!\!\$\ \mathsf{Maul}_1(x_1, \alpha_1, \rho_1)$		
			$\xrightarrow{\widehat{\alpha}_0, \widehat{\alpha}_1}$	
				$\beta \leftarrow\!\!\$\ \{0,1\}^\ell$
			$\xleftarrow{\beta}$	
		$\rho = \rho_0 \oplus \rho_1$ $\widehat{\beta} = \beta \oplus \rho$		
	$\xleftarrow{\widehat{\beta}}$			
$\beta_b = \widehat{\beta} \oplus \beta_{1-b}$ $\gamma_b = \mathsf{P}_2^b(x_b, w, \beta_b, a)$				
	$\xrightarrow{\gamma_0, \gamma_1, \beta_0, \beta_1}$			
		$\widehat{\beta}_0 = \beta_0 \oplus \rho_0$ $\widehat{\beta}_1 = \beta_1 \oplus \rho_1$ $\widehat{\gamma}_0 = \mathsf{Bal}_0(\gamma_0, \sigma_0)$ $\widehat{\gamma}_1 = \mathsf{Bal}_1(\gamma_1, \sigma_1)$		
			$\xrightarrow{\widehat{\gamma}_0, \widehat{\gamma}_1, \widehat{\beta}_0, \widehat{\beta}_1}$	
				$\beta \stackrel{?}{=} \widehat{\beta}_0 \oplus \widehat{\beta}_1$ $\mathsf{V}^0(x_0, (\widehat{\alpha}_0, \widehat{\beta}_0, \widehat{\gamma}_0)) \stackrel{?}{=} 1$ $\mathsf{V}^1(x_1, (\widehat{\alpha}_1, \widehat{\beta}_1, \widehat{\gamma}_1)) \stackrel{?}{=} 1$

Figure 6 Reverse Firewall for the basic OR composition of Sigma protocols, where $b \in \{0,1\}$ is s.t. $(x_b, w) \in \mathcal{R}_b$.

▶ **Theorem 6.** *Let $\Sigma_0 = (\mathsf{P}^0 = (\mathsf{P}_1^0, \mathsf{P}_2^0), \mathsf{V}^0)$ and $\Sigma_1 = (\mathsf{P}^1 = (\mathsf{P}_1^1, \mathsf{P}_2^1), \mathsf{V}^1)$ be instance-dependent malleable Sigma protocols with unique responses, for relations $\mathcal{R}_0$ and $\mathcal{R}_1$. The RF W of Fig. 6 preserves completeness, and is weakly HVZK/WI preserving for the prover of the Sigma protocol Σ_{OR} for relation $\mathcal{R}_{\mathsf{OR}}$.*

5 Previous Work

5.1 Comparison with Mironov and Stephens-Davidowitz

In their original paper, Mironov and Stephens-Davidowitz [40] build RFs for arbitrary two-party protocols. While their results are related to ours, since IPSes are just a special case of two-party computation, there are some crucial differences which we highlight below.

The first RF construction sanitizes a specific combination of re-randomizable garbled circuits and oblivious transfer, for obtaining general-purpose private function evaluation. The second RF construction sanitizes any two-party protocol, at the price of encrypting the full transcript under public keys that are broadcast at the beginning of the protocol. Both constructions can be instantiated based on (variants of) the DDH assumption. When cast to IPSes, their results yield:

(i) A RF for the prover that weakly preserves ZK. This is comparable to our RF achieving weak ZK preservation using malleable Sigma protocols and key-malleable commitments. However, our constructions have the advantage that we do not need to change the initial IPS, and thus our RF can be applied directly to already existing implementations in a fully transparent manner (and without introducing any overhead).

(ii) A RF for the prover satisfying a property called strong exfiltration resistance *against an eavesdropper*, which means that exfiltration resistance holds w.r.t. an arbitrarily subverted prover talking to the *honest verifier*. Note that the latter does not contradict our impossibility result ruling out strong ZK preservation, as our attacks crucially rely on the fact that the distinguisher can (passively) corrupt the verifier.

(iii) A RF for the verifier satisfying both strong exfiltration resistance and the following weak guarantee: No malicious prover can find statements x_0, x_1 such that it can distinguish transcripts obtained by talking to an arbitrarily subverted verifier holding either input x_0 or input x_1. Note that the latter does not contradict our impossibility result that rules out weak soundness preservation, since none of the above guarantees imply soundness preservation.

We observe that the above results have at least one of the following drawbacks: (i) The RF is not transparent, *i.e.* it cannot be applied to the initial protocol as is; (ii) The resulting sanitized protocol is not efficient, as we first need to encode the function being computed as a circuit.

Our techniques allow to overcome these limitations in the concrete case of IPSes, as our RFs are both transparent (*i.e.* they can be applied directly to already deployed protocols) and efficient (*i.e.* the sanitized IPSes have exactly the same efficiency as the original, both in terms of round and communication complexity). We see this as the main novelty of our work.

5.2 Additional Related Works

Besides the already mentioned constructions, RFs have also been realized in other settings including digital signatures [5], secure message transmission and key exchange [21, 12], and oblivious transfer [40, 12].

Moreover, a few other lines of research recently[10] emerged to tackle the challenge of protecting cryptographic algorithms against (different forms of) subversion. We review the

[10] All these research directions have their roots in the seminal works of Young and Yung [52] and Simmons [50], in the settings of kleptography and subliminal channels.

main ones below.

Algorithm substitution attacks

Bellare, Patterson, and Rogaway [11] studied subversion of symmetric encryption schemes in the form of algorithm substitution attacks (ASAs). In particular, they show that *undetectable* subversion of the encryption algorithm is possible, and may lead to severe security breaches; moreover, they prove that deterministic, stateful, ciphers are secure against the same type of ASAs. Follow-up works improved the original paper in several aspects [18, 10], and explored the power of ASAs in other contexts, *e.g.* digital signatures [5], secret sharing [31], and message authentication codes [3].

Backdoors

Another form of subversion consists of all those attacks that surreptitiously generate public parameters (primes, curves, etc.) together with secret backdoors that allow to bypass security. The study of this type of subversion is motivated by the DUAL_EC_DRBG PRG incident.

A formal study of parameters subversion has been considered for several primitives, including pseudorandom generators [20, 19], hash functions [27], non-interactive zero knowledge [8], and public-key encryption [6].

Cliptography

Russell *et al.* [46] (see also [47, 4]) consider a different approach to the immunization of cryptosystems against complete subversion (*i.e.*, when all algorithms can be subverted by the attacker): offline/online black-box testing. This amounts to introducing an external entity, called the watchdog, whose goal is to test, either in an online or in an offline fashion, whether a given cryptographic implementation is compliant with its specification.

Hence, a cryptosystem is deemed secure against complete subversion if there exists a universal watchdog such that, for every attacker subverting all algorithms, either the watchdog detects subversion with high probability, or the cryptoscheme remains secure even when using its subverted implementation.

Self-guarding

Yet another approach towards thwarting subversion is that of self-guarding [28]. The idea here is to assume a trusted initialization phase in which the honest parties possess a genuine implementation of the cryptosystem, before subversion takes place. This phase is used in order to generate samples that will be exploited later, together with additional simple operations that need to be implemented from scratch, to prevent leakage in the face of subversion attacks.

6 Conclusion

We showed how to design cryptographic reverse firewalls allowing to preserve security of interactive proof systems in the face of subversion. Our firewalls apply to a large class of Sigma protocols meeting a natural malleability property, and can be extended to cover classical applications of Sigma protocols for designing zero-knowledge proofs and for proving compound statements.

We leave it as an intriguing open problem to design a reverse firewall for the OR composition of Sigma protocols that are delayed input, as considered in [13, 14].

References

1. Vulnerability summary for cve-2014-6271 (shellshock), September 2014. URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271.
2. Juniper vulnerability, 2015. URL: https://kb.juniper.net/InfoCenter/index?page=content&id=JSA10713.
3. Marcel Armour and Bertram Poettering. Substitution attacks against message authentication. *IACR Trans. Symmetric Cryptol.*, 2019(3):152–168, 2019.
4. Giuseppe Ateniese, Danilo Francati, Bernardo Magri, and Daniele Venturi. Public immunization against complete subversion without random oracles. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *ACNS 19*, volume 11464 of *LNCS*, pages 465–485. Springer, Heidelberg, June 2019. doi:10.1007/978-3-030-21568-2_23.
5. Giuseppe Ateniese, Bernardo Magri, and Daniele Venturi. Subversion-resilient signature schemes. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS 2015*, pages 364–375. ACM Press, October 2015. doi:10.1145/2810103.2813635.
6. Benedikt Auerbach, Mihir Bellare, and Eike Kiltz. Public-key encryption resistant to parameter subversion and its realization from efficiently-embeddable groups. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part I*, volume 10769 of *LNCS*, pages 348–377. Springer, Heidelberg, March 2018. doi:10.1007/978-3-319-76578-5_12.
7. James Ball, Julian Borger, and Glenn Greenwald. Revealed: How US and UK spy agencies defeat internet privacy and security. *Guardian Weekly*, September 2013.
8. Mihir Bellare, Georg Fuchsbauer, and Alessandra Scafuro. NIZKs with an untrusted CRS: Security in the face of parameter subversion. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 777–804. Springer, Heidelberg, December 2016. doi:10.1007/978-3-662-53890-6_26.
9. Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 390–420. Springer, Heidelberg, August 1993. doi:10.1007/3-540-48071-4_28.
10. Mihir Bellare, Joseph Jaeger, and Daniel Kane. Mass-surveillance without the state: Strongly undetectable algorithm-substitution attacks. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS 2015*, pages 1431–1440. ACM Press, October 2015. doi:10.1145/2810103.2813681.
11. Mihir Bellare, Kenneth G. Paterson, and Phillip Rogaway. Security of symmetric encryption against mass surveillance. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 1–19. Springer, Heidelberg, August 2014. doi:10.1007/978-3-662-44371-2_1.
12. Rongmao Chen, Yi Mu, Guomin Yang, Willy Susilo, Fuchun Guo, and Mingwu Zhang. Cryptographic reverse firewall via malleable smooth projective hash functions. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 844–876. Springer, Heidelberg, December 2016. doi:10.1007/978-3-662-53887-6_31.
13. Michele Ciampi, Giuseppe Persiano, Alessandra Scafuro, Luisa Siniscalchi, and Ivan Visconti. Improved OR-composition of sigma-protocols. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 112–141. Springer, Heidelberg, January 2016. doi:10.1007/978-3-662-49099-0_5.
14. Michele Ciampi, Giuseppe Persiano, Alessandra Scafuro, Luisa Siniscalchi, and Ivan Visconti. Online/offline OR composition of sigma protocols. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 63–92. Springer, Heidelberg, May 2016. doi:10.1007/978-3-662-49896-5_3.
15. Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 174–187. Springer, Heidelberg, August 1994. doi:10.1007/3-540-48658-5_19.

16 Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally efficient multi-authority election scheme. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 103–118. Springer, Heidelberg, May 1997. doi:10.1007/3-540-69053-0_9.

17 Ivan Damgård and Jens Groth. Non-interactive and reusable non-malleable commitment schemes. In *35th ACM STOC*, pages 426–437. ACM Press, June 2003. doi:10.1145/780542.780605.

18 Jean Paul Degabriele, Pooya Farshim, and Bertram Poettering. A more cautious approach to security against mass surveillance. In Gregor Leander, editor, *FSE 2015*, volume 9054 of *LNCS*, pages 579–598. Springer, Heidelberg, March 2015. doi:10.1007/978-3-662-48116-5_28.

19 Jean Paul Degabriele, Kenneth G. Paterson, Jacob C. N. Schuldt, and Joanne Woodage. Backdoors in pseudorandom number generators: Possibility and impossibility results. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 403–432. Springer, Heidelberg, August 2016. doi:10.1007/978-3-662-53018-4_15.

20 Yevgeniy Dodis, Chaya Ganesh, Alexander Golovnev, Ari Juels, and Thomas Ristenpart. A formal treatment of backdoored pseudorandom generators. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 101–126. Springer, Heidelberg, April 2015. doi:10.1007/978-3-662-46800-5_5.

21 Yevgeniy Dodis, Ilya Mironov, and Noah Stephens-Davidowitz. Message transmission with reverse firewalls—secure communication on corrupted machines. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 341–372. Springer, Heidelberg, August 2016. doi:10.1007/978-3-662-53018-4_13.

22 Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. On the non-malleability of the Fiat-Shamir transform. In Steven D. Galbraith and Mridul Nandi, editors, *INDOCRYPT 2012*, volume 7668 of *LNCS*, pages 60–79. Springer, Heidelberg, December 2012. doi:10.1007/978-3-642-34931-7_5.

23 Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st FOCS*, pages 308–317. IEEE Computer Society Press, October 1990. doi:10.1109/FSCS.1990.89549.

24 Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *22nd ACM STOC*, pages 416–426. ACM Press, May 1990. doi:10.1145/100216.100272.

25 Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987. doi:10.1007/3-540-47721-7_12.

26 Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 152–168. Springer, Heidelberg, August 2005. doi:10.1007/11535218_10.

27 Marc Fischlin, Christian Janson, and Sogol Mazaheri. Backdoored hash functions: Immunizing HMAC and HKDF. In *31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018*, pages 105–118, 2018.

28 Marc Fischlin and Sogol Mazaheri. Self-guarding cryptographic protocols against algorithm substitution attacks. In *31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018*, pages 76–90, 2018.

29 Chaya Ganesh, Bernardo Magri, and Daniele Venturi. Cryptographic reverse firewalls for interactive proof systems. *IACR Cryptology ePrint Archive*, 2020:204, 2020. URL: https://eprint.iacr.org/2020/204.

30 Juan A. Garay, Philip D. MacKenzie, and Ke Yang. Strengthening zero-knowledge protocols using signatures. *Journal of Cryptology*, 19(2):169–209, April 2006. doi:10.1007/s00145-005-0307-3.

31 Irene Giacomelli, Ruxandra F. Olimid, and Samuel Ranellucci. Security of linear secret-sharing schemes against mass surveillance. In Michael Reiter and David Naccache, editors, *CANS 15*, LNCS, pages 43–58. Springer, Heidelberg, December 2015. doi:10.1007/978-3-319-26823-1_4.

32 Oded Goldreich. *Foundations of Cryptography: Basic Tools*, volume 1. Cambridge University Press, Cambridge, UK, 2001.

33 Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology*, 9(3):167–190, June 1996.

34 Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):691–729, 1991.

35 Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *14th ACM STOC*, pages 365–377. ACM Press, May 1982. doi:10.1145/800070.802212.

36 Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

37 Louis C. Guillou and Jean-Jacques Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both trasmission and memory. In C. G. Günther, editor, *EUROCRYPT'88*, volume 330 of *LNCS*, pages 123–128. Springer, Heidelberg, May 1988. doi:10.1007/3-540-45961-8_11.

38 Arjen K. Lenstra, James P. Hughes, Maxime Augier, Joppe W. Bos, Thorsten Kleinjung, and Christophe Wachter. Public keys. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 626–642. Springer, Heidelberg, August 2012. doi:10.1007/978-3-642-32009-5_37.

39 Ueli M. Maurer. Unifying zero-knowledge proofs of knowledge. In Bart Preneel, editor, *AFRICACRYPT 09*, volume 5580 of *LNCS*, pages 272–286. Springer, Heidelberg, June 2009.

40 Ilya Mironov and Noah Stephens-Davidowitz. Cryptographic reverse firewalls. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 657–686. Springer, Heidelberg, April 2015. doi:10.1007/978-3-662-46803-6_22.

41 Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 31–53. Springer, Heidelberg, August 1993. doi:10.1007/3-540-48071-4_3.

42 Tatsuaki Okamoto and Kazuo Ohta. Divertible zero knowledge interactive proofs and commutative random self-reducibility. In Jean-Jacques Quisquater and Joos Vandewalle, editors, *EUROCRYPT'89*, volume 434 of *LNCS*, pages 134–148. Springer, Heidelberg, April 1990. doi:10.1007/3-540-46885-4_16.

43 Rafail Ostrovsky, Vanishree Rao, and Ivan Visconti. On selective-opening attacks against encryption schemes. In Michel Abdalla and Roberto De Prisco, editors, *SCN 14*, volume 8642 of *LNCS*, pages 578–597. Springer, Heidelberg, September 2014. doi:10.1007/978-3-319-10879-7_33.

44 Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 129–140. Springer, Heidelberg, August 1992. doi:10.1007/3-540-46766-1_9.

45 Nicole Perlroth, Jeff Larson, and Scott Shane. N.S.A. able to foil basic safeguards of privacy on web. *The New York Times*, September 2013.

46 Alexander Russell, Qiang Tang, Moti Yung, and Hong-Sheng Zhou. Cliptography: Clipping the power of kleptographic attacks. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 34–64. Springer, Heidelberg, December 2016. doi:10.1007/978-3-662-53890-6_2.

47 Alexander Russell, Qiang Tang, Moti Yung, and Hong-Sheng Zhou. Generic semantic security against a kleptographic adversary. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 907–922. ACM Press, October / November 2017. doi:10.1145/3133956.3133993.

48 Alessandra Scafuro and Ivan Visconti. On round-optimal zero knowledge in the bare public-key model. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 153–171. Springer, Heidelberg, April 2012. doi:10.1007/978-3-642-29011-4_11.

49 Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, Heidelberg, August 1990. doi:10.1007/0-387-34805-0_22.

50 Gustavus J. Simmons. The prisoners' problem and the subliminal channel. In David Chaum, editor, *CRYPTO'83*, pages 51–67. Plenum Press, New York, USA, 1983.

51 Dominique Unruh. Quantum proofs of knowledge. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 135–152. Springer, Heidelberg, April 2012. doi:10.1007/978-3-642-29011-4_10.

52 Adam Young and Moti Yung. Kleptography: Using cryptography against cryptography. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 62–74. Springer, Heidelberg, May 1997. doi:10.1007/3-540-69053-0_6.