Information and Computation 247 (2016) 1–10

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Combined schemes for signature and encryption: The public-key and the identity-based setting

María Isabel González Vasco [a,*], Florian Hess [b,*], Rainer Steinwandt [c,*]

[a] Departamento de Matemática Aplicada, Universidad Rey Juan Carlos, c/ Tulipán, s/n, 28933 Madrid, Spain
[b] Institut für Mathematik, Carl von Ossietzky Universität, 26111 Oldenburg, Germany
[c] Department of Mathematical Sciences, Florida Atlantic University, 777 Glades Road, Boca Raton, FL 33431, USA

ARTICLE INFO

Article history:
Received 3 November 2008
Received in revised form 15 August 2013
Available online 1 December 2015

Keywords:
Combined scheme
Identity-based cryptography
Public-key cryptography
Key separation

ABSTRACT

Consider a scenario in which parties use a public-key encryption scheme and a signature scheme with a single public key/private key pair—so the private key *sk* is used for both signing and decrypting. Such a simultaneous use of a key is in general considered poor cryptographic practice, but from an efficiency point of view looks attractive.
We offer security notions to analyze such violations of key separation. For both the identity- and the non-identity-based setting, we show that—although being insecure in general—for schemes of interest the resulting *combined scheme* can offer strong security guarantees.

1. Introduction

Using a single cryptographic key for different purposes is commonly considered poor cryptographic practice, as it violates the design principle of key separation. Notwithstanding this, already in the late 90s Kelsey et al. [18] noted that there exist *forces pushing us toward a world in which different applications share common key material*: avoiding the cost for multiple certificates, (non-cryptographic) applications that simply default to a single user-specific key, and resource limitations on smart cards. For typical signature and public-key encryption schemes it may well happen that the secret-key-dependent operations are the very same—e.g., an exponentiation in a suitable group. If costly protection measures against side-channel or fault induction attacks need to be implemented, it is particularly tempting to work with a single key pair. Provided that there are no accidental interactions (in the sense of [18, Section 3]), one may hope for synergies in code size and implementation cost.

Haber and Pinkas [16] show that the simultaneous use of related keys in a signature scheme and a public-key encryption scheme is, for several examples, secure in a strong sense. They consider an adversary against a signature scheme which has unrestricted access to a decryption oracle of an encryption scheme using a related secret key, and prove that for several signature schemes such adversaries are not more damaging than “standard” ones. Analogously, for some encryption schemes, they prove that an attacker who is granted unrestricted access to a signing oracle of a signature scheme using a related secret key will not endanger the security of the encryption scheme. Subsequent work focused on *universal padding schemes* that can be used for both signing and encryption without the need of separate keys. Coron et al. showed that PSS enables such a secure composition of a signature and encryption scheme with a single key pair [10]. More recently, Komano and Ohta

* Corresponding author.
E-mail addresses: mariaisabel.vasco@urjc.es (M.I. González Vasco), florian.hess@uni-oldenburg.de (F. Hess), rsteinwa@fau.edu (R. Steinwandt).

http://dx.doi.org/10.1016/j.ic.2015.11.001
0890-5401/

proposes combined constructions building on OAEP+ and REACT. Instead of the *partial-domain one-wayness* requirement of Coron et al., [19] imposes a one-wayness requirement only. Further refinements of universal paddings are explored by Chevallier-Mames et al. in [9].

Our contribution Section 2 follows Haber and Pinkas [16] in the sense that we try to combine existing schemes that have not been designed for usage with a common private key. We analyze the security of such *combined schemes* using dedicated security notions building on the ones coined by Komano and Ohta in [19]. After showing how the simultaneous use of a private key can be fatal, we give a *combined scheme* with a security proof. This is constructed from the ElGamal signature scheme in the modification of Pointcheval and Stern [22] and an ElGamal encryption scheme under a Fujisaki–Okamoto conversion. We prove the resulting scheme to be secure in a strong sense: in the random oracle model, both existential unforgeability and indistinguishability of encryptions are achieved.

In the identity-based setting, working with a single user identity and one corresponding user key appears particularly natural, and Section 3 explores (for the first time in this context) the use of a unique private key in an identity-based setting: an identity-based encryption scheme and an identity-based signature scheme share a setup and key extraction algorithm and each user has one secret key only which is used for both signing and decrypting. We prove that such a simultaneous use can be possible without jeopardizing the security of the involved schemes. Namely, for an identity based signature scheme by Hess [17] and an identity based encryption scheme of Boneh and Franklin [8] we prove security in the sense of a natural generalization of standard security notions in identity-based cryptography.

Related (follow-up) work In the years after making a preprint of our results available [23], some related work has appeared: the work of Degabriele et al. [11] on the EMV standards shows that EMV's RSA-based algorithms have security problems if a single key-pair is used for both signature and encryption; on the other hand, the elliptic curve algorithms that may end up as part of these standards are shown to be secure. Furthermore, in [20] Paterson et al. provide a way to construct a combined public-key scheme by means of an identity-based encryption scheme. They also offer a more efficient technique to obtain a combined public-key scheme, using the signature scheme of Boneh and Boyen [5] and an identity-based encryption scheme by the same authors [6].

If the essential application of an encryption and a signature scheme in a protocol consists of signing messages with a sender's private key followed by encrypting the signed messages under a recipient's public key, then signcryption [24] can be an alternative to separate encryption and signature mechanisms. As detailed in [1], a signcryption scheme induces a signature and an encryption scheme. With regard to key lengths, however, these induced schemes appear inferior to dedicated encryption or signature mechanisms, as essentially two signcryption keys are used to form one key for the induced signature or encryption scheme. For a scenario where we want the flexibility of separate encryption and signature mechanisms, the use of a signcryption scheme appears less attractive than a "secure key reuse" as described below. To find analogues to our security goals one would actually look at insider security against multi-user signcryption [1,12] where both indistinguishability of ciphertexts and existential unforgeability must be achieved. More recently, Arriaga et al. [2] discussed *randomness reuse* when dealing with signcryption; their encrypt-then-sign and sign-then-encrypt constructions with randomness reuse are somewhat dual to the key reuse we consider. There, the key generations for the invoked encryption and signature schemes are independent, but the random coins used in the computation of a ciphertext and a signature coincide.

2. Combined public-key schemes

2.1. Preliminaries and definitions

Adapting the terminology from [16], we define a *combined public-key scheme* as a combination of a public-key encryption scheme and a signature scheme that have the key generation in common:

Definition 1 *(Combined public-key scheme).* A *combined public-key scheme* is a tuple $(\mathcal{K}, \mathcal{E}, \mathcal{D}, \mathcal{S}, \mathcal{V})$ of polynomial time algorithms:

- $\mathcal{K}$ is a probabilistic *key generation algorithm* that on input the security parameter 1^k outputs a public key/secret key pair (pk, sk).
- $\mathcal{E}$ is a probabilistic *encryption algorithm* that on input a message m and a public key pk computes a ciphertext $c \leftarrow \mathcal{E}_{pk}(m)$.
- $\mathcal{D}$ is a deterministic *decryption algorithm* that on input a candidate ciphertext c and a secret key sk outputs a plaintext $m \leftarrow \mathcal{D}_{sk}(c)$ or an error symbol $\perp$.
- $\mathcal{S}$ is a probabilistic *signing algorithm* that on input a message m and a secret key sk outputs a signature $\sigma \leftarrow \mathcal{S}_{sk}(m)$.
- $\mathcal{V}$ is a deterministic *verification algorithm* that on input a public key pk, a message m and a candidate signature σ outputs true or false.

For a pair (pk, sk) generated by $\mathcal{K}$ we require that with overwhelming probability the obvious correctness condition holds: For all messages m we have $\mathcal{D}_{sk}(\mathcal{E}_{pk}(m)) = m$ and $\mathcal{V}_{pk}(m, \mathcal{S}_{sk}(m)) = \text{true}$.