Unifying Kleptographic Attacks

George Teşeleanu[1,2]

[1] Advanced Technologies Institute
10 Dinu Vintilă, Bucharest, Romania
tgeorge@dcti.ro
[2] Department of Computer Science
"Al.I.Cuza" University of Iaşi 700506 Iaşi, Romania,
george.teseleanu@info.uaic.ro

Abstract. We present two simple backdoors that can be implemented into Maurer's unified zero-knowledge protocol. Thus, we show that a high level abstraction can replace individual backdoors embedded into protocols for proving knowledge of a discrete logarithm (*e.g.* the Schnorr and Girault protocols), protocols for proving knowledge of an e^{th}-root (*e.g.* the Fiat-Shamir and Guillou-Quisquater protocols), protocols for proving knowledge of a discrete logarithm representation (*e.g.* the Okamoto protocol) and protocols for proving knowledge of an e^{th}-root representation.

1 Introduction

Classical security models assume that the cryptographic algorithms found in a device are correctly implemented and according to technical specifications. Unfortunately, in the real world, users have little control over the design criteria or the implementation of a security module. When using a hardware device, for example a smartcard, the user implicitly assumes an honest manufacturer that builds devices according to the provided specifications. The idea of a malicious manufacturer that tampers with the device or embeds a backdoor in an implementation was first suggested by Young and Yung [32, 33]. As proof of concept, they developed secretly embedded trapdoor with universal protection (SETUP) attacks.

Although considered far-fetched by some cryptographers, SETUP[3] attacks were found in real world implementations [9, 10]. These attacks are based on the usage of the Dual-EC generator, a cryptographically secure pseudorandom number generator standardized by NIST. Internal NSA documents leaked by Edward Snowden [3, 26] indicated a backdoor embedded into the Dual-EC generator. Shortly afterward, the aforementioned examples were found. This backdoor is a direct application of the work conducted by Young and Yung [32–35].

A consequence of Snowden's revelations is the revival of this research area [2, 4, 7, 12, 16, 21, 27, 28, 31]. In [5], SETUP attacks applied to symmetric encryption schemes are re-branded as *algorithmic substitution attacks* (ASA). A link between *secret-key steganography* and ASAs can be found in [7]. More generic attacks (*subversion attacks*) tailored for signature schemes are introduced in [2]. Subversion attacks include SETUP attacks and ASAs, but generic malware and virus attacks are also included. Generic protections against backdoored PRNGs, such as the Dual-EC generator, are studied in [27, 28].

The initial model proposed by Young and Yung is the black-box model[4]. For our intended purposes this model suffices, since the zero-knowledge protocols we attack were designed for smartcards. Note that even if we relax this model and assume that the code is open-source, according to [5], the sheer complexity of open-source software and the small number of experts who review them still make ASAs plausible. Note that these attacks need a malicious device manufacturer[5] to work. An important property is that infected smartcards should have inputs and outputs indistinguishable from regular smartcards. However, if the smartcard is reverse engineered, the deployed mechanism may be detectable.

[3] secretly embedded trapdoor with universal protection

[4] A black-box is a device, process or system, whose inputs and outputs are known, but its internal structure or working is not known or accessible to the user (*e.g.* tamper proof devices).

[5] that implements the mechanisms to recover the keys

There are two methods to embed backdoors into a system: either you generate special public parameters (SPP) or you infect the random numbers (IRN) used by the system. In the case of discrete logarithm based systems, SPP and IRN were studied in [16,19,21,31–35]. We only found SPP [11,32,33,35,36] and not IRN in the case of factorization based systems.

Zero-knowledge protocols were introduced as a mean to prove one's identity. These protocols are defined between a prover (usually called *Peggy*) that possesses some secret x[6] and a verifier (usually called $Victor$) that checks if *Peggy* really possesses x. Two classical examples of such protocols are the Schnorr protocol [29] and the Guillou-Quisquater protocol [20]. Note that both protocols were proposed for smartcards. By abstracting the two protocols, Maurer shows [22] that they are actually instantiations of the same protocol.

Using the same level of abstraction as in [22], we show how an attacker (called $Mallory$) can mount a SETUP attack and extract *Peggy*'s secret. When instantiated, this attack provides new insight into SETUP attacks. In particular, we provide the first IRN attack on a factoring based system and the first attack on systems based on e^{th}-root representations[7]. We also provide the reader with new instantiations of Maurer's unified protocol: the Girault protocol, a new proof of knowledge for discrete logarithm representation in $\mathbb{Z}_n^*$ and a proof of knowledge of an e^{th}-root representation.

The second SETUP attack we introduce is a generalization of Young and Yung's work. When instantiated with the Schnorr protocol, we obtain their results. We also provide other examples not mentioned by Young and Yung.

Structure of the paper. We introduce notations and definitions used throughout the paper in Section 2. In Section 3 we present our new general SETUP attacks and prove them secure. Instantiations of our attacks can be found in Section 4. We conclude in Section 5. Additional definitions are given in Appendix A.

2 Preliminaries

Notations. Throughout the paper, the notation $|S|$ denotes the cardinality of a set S. The action of selecting a random element x from a sample space X is denoted by $x \xleftarrow{\$} X$, while $x \leftarrow y$ represents the assignment of value y to variable x. The probability of the event E to happen is denoted by $Pr[E]$. The subset $\{0, \ldots, s\} \in \mathbb{N}$ is denoted by $[0, s]$.

2.1 Groups

Let $(\mathbb{G}, \star)$ and $(\mathbb{H}, \otimes)$ be two groups. We assume that the group operations $\star$ and $\otimes$ are efficiently computable. Compared to [22], we also assume that $\mathbb{G}$ is a cyclic group. Note that this implies that $\mathbb{G}$ is commutative. Let g be a generator of $\mathbb{G}$. We denote by αg the element $g \star \ldots \star g$ obtained by repeatedly applying the group operation $\alpha - 1$ times.

Let $f : \mathbb{G} \to \mathbb{H}$ be a function (not necessarily one-to-one). We say that f is a homomorphism if $f(x \star y) = f(x) \otimes f(y)$. Throughout the paper we consider f to be a one-way function, *i.e.* it is infeasible to compute x from $f(x)$. To be consistent with [22], we denote by $[x]$ the value $f(x)$. Note that given $[x]$ and $[y]$ we can efficiently compute $[x \star y] = [x] \otimes [y]$, due to the homomorphism. By $[g]^\alpha$ we denote $[\alpha g] = [g] \otimes \ldots \otimes [g]$ (α times).

Definition 1 (Hash Diffie-Hellman - HDH). *Let $\mathbb{D}$ be a cyclic group of order q, d a generator of $\mathbb{D}$, $\mathbb{E}$ a group and $h : \mathbb{D} \to \mathbb{E}$ a hash function. Let A be a PPT algorithm which returns 1 on input (d^x, d^y, z) if $h(d^{xy}) = z$. We define the advantage*

$$ADV^{HDH}_{\mathbb{D},d,h}(A) = |Pr[A(d^x, d^y, h(d^{xy})) = 1 | x, y \xleftarrow{\$} \mathbb{Z}_q^*] - Pr[A(d^x, d^y, z) = 1 | x, y \xleftarrow{\$} \mathbb{Z}_q^*, z \xleftarrow{\$} \mathbb{E}]|.$$

If $ADV^{HDH}_{\mathbb{D},d,h}(A)$ is negligible for any PPT algorithm A, we say that the Hash Diffie-Hellman problem is hard in $\mathbb{D}$.

[6] associated with her identity

[7] For systems based on discrete logarithm representations a backdoor was described in [31].

Remark 1. According to [6], the HDH assumption is equivalent with the computational Diffie-Hellman (CDH) assumption[8] in the random oracle model. If the decisional Diffie-Hellman (DDH) assumption[8] is hard in $\mathbb{D}$ and h is entropy smoothing[8], then the HDH assumption is hard in $\mathbb{D}$ [1,24,30]. In [17], the authors show that the HDH assumption holds, even if the DDH assumption is relaxed to the following assumption: $\mathbb{D}$ contains a large enough group in which DDH holds. A particularly interesting group is $\mathbb{Z}_p^*$, where p is a "large"[9] prime. According to [17], it is conjectured that if $\mathbb{D}$ is generated by an element $d \in \mathbb{Z}_p^*$ of order q, where q is a "large"[10] prime that divides $p-1$, then the DDH assumption holds. The analysis conducted in [17] provides the reader with solid arguments to support the hypothesis that HDH holds in the subgroup $\mathbb{D} \subset \mathbb{Z}_p^*$.

2.2 Zero-Knowledge Protocols

Let $Q : \{0,1\}^* \times \{0,1\}^* \to \{\texttt{true}, \texttt{false}\}$ be a predicate. Given a value z, Peggy will try to convince Victor that she knows a value x such that $Q(z,x) = \texttt{true}$. We further recall a definition from [14] that captures the notion that being successful in a protocol (P,V) implies knowledge of a value x such that $Q(z,x) = \texttt{true}$.

Definition 2 (Proof of Knowledge Protocol). *An interactive protocol (P,V) is a proof of knowledge protocol for predicate Q if the following properties hold*

- *Completeness: V accepts the proof when P has as input an x with $Q(z,x) = \texttt{true}$;*
- *Soundness: there is an efficient program K (called knowledge extractor) such that for any $\hat{P}$ (possibly dishonest) with non-negligible probability of making V accept the proof, K can interact with $\hat{P}$ and output (with overwhelming probability) an x such that $Q(z,x) = \texttt{true}$.*

Definition 3 (2-extractable). *Let Q be a predicate for a proof of knowledge. A 3-move protocol[11] with challenge space $\mathcal{C}$ is 2-extractable if from any two triplets (t,c,r) and (t,c',r'), with distinct $c,c' \in \mathcal{C}$ accepted by Victor, one can efficiently compute an x such that $Q(z,x) = \texttt{true}$.*

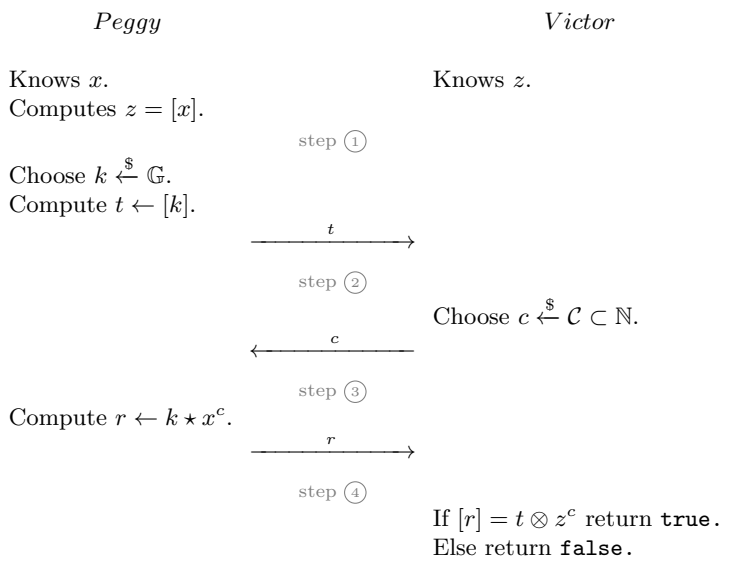

Fig. 1. Maurer's Unified Zero-Knowledge (UZK) Protocol.

[8] We refer the reader to Appendix A for a definition of the concept.
[9] at least 2048 bits, better 3072 bits
[10] at least 192 bits, better 256 bits
[11] *Peggy* sends t, *Victor* sends c, *Peggy* sends r

According to [22], UZK (Figure 1) is a zero-knowledge protocol if the conditions from Theorem 1 are satisfied. If the challenge space $\mathcal{C}$ is small, then one needs several 3-move rounds to make the soundness error negligible.

Theorem 1. *If values $\ell \in \mathbb{Z}$ and $u \in \mathbb{G}$ are known such that*

- $\gcd(c_0 - c_1, \ell) = 1$ *for all* $c_0, c_1 \in \mathcal{C}$ *with* $c_0 \neq c_1$,
- $[u] = z^{\ell}$,

then the protocol described in Figure 1 is 2-extractable. Moreover, a protocol consisting of s rounds is a proof of knowledge if $1/|\mathcal{C}|^s$ is negligible, and it is a zero-knowledge protocol if $|\mathcal{C}|$ is polynomially bounded.

2.3 SETUP attacks

Definition 4 (Secretly Embedded Trapdoor with Universal Protection - SETUP**).** *A Secretly Embedded Trapdoor with Universal Protection (SETUP) is an algorithm that can be inserted in a system such that it leaks encrypted private key information to an attacker through the system's outputs. Encryption of the private key is performed using an asymmetric encryption scheme. It is assumed that the decryption function is accessible only to the attacker.*

Definition 5 (SETUP **indistinguishability -** IND-SETUP**).** *Let C_0 be a black-box system that uses a secret key sk. Let $\mathcal{AE}$ be the asymmetric encryption scheme used by a SETUP mechanism as defined above, in Definition 4. We consider C_1 an altered version of C_0 that contains a SETUP mechanism based on $\mathcal{AE}$. Let A be a PPT algorithm which returns 1 if it detects that C_0 is altered. We define the advantage*

$$ADV^{\text{IND-SETUP}}_{C_0,C_1}(A) = |Pr[A^{C_1(sk,\cdot)}(\lambda) = 1] - Pr[A^{C_0(sk,\cdot)}(\lambda) = 1]|.$$

If $ADV^{\text{IND-SETUP}}_{C_0,C_1}(A)$ is negligible for any PPT algorithm A, we say that C_0 and C_1 are polynomially indistinguishable.

Remark 2. Definition 5 is a formalization of the indistinguishability property for a regular SETUP mechanism described in [33]. The authors of [2] propose a more general concept (*public undetectability*) that allows *Mallory* to tailor his attacks depending on each of his victim's public key. The two formalizations, SETUP indistinguishability and public undetectability, assume that the public parameters $(g, \mathbb{G}, \mathbb{H})$ and the secret/public key pair (x, z) are honestly generated. In some cases, *Mallory* can also maliciously generate these. This scenario is captured in [27] (*cliptographic game*). A consequence of the three formalizations is that C_0 and C_1 have the same security.

Remark 3. We consider that the attacks presented from now on are implemented in a device D that is used by *Peggy* to prove the knowledge of x. We assume that x is stored only in D's volatile memory[12]. Note that *Peggy* believes that D works in accordance with the UZK protocol.

Remark 4. UZK can be transformed into a signature scheme using the Fiat-Shamir transform [15]. Thus, obtaining a unified signature scheme. Note that the SETUP attacks described for UZK are preserved by the Fiat-Shamir transform, therefore *Mallory* can recover *Peggy*'s signing key by using either of them.

3 Unified SETUP Attacks

In this section we state the principal results of this paper. The main protocol is a SETUP attack against UZK that allows *Mallory* to extract *Peggy*'s knowledge of x, while the supplementary one only allows *Mallory*

[12] If *Peggy* knows her secret she is able to detect the SETUP mechanism using its description and parameters (found by means of reverse engineering a black-box, for example).

to compute x in some specific instantiations of UZK. We only show how to infect two sessions of the protocol and assume that the rest of the sessions remain unmodified.

Before stating the results, we first make some preliminary assumptions. Let $h : \mathbb{H} \to \mathbb{G}$ be a hash function and let $i = 0, 1$ be an index. We assume that *Peggy* runs the protocols at least two times (*i.e.* once for $i = 0$ and once for $i = 1$). We denote by $y \leftarrow [g]^{x_M}$ *Mallory*'s public key, while $x_M \xleftarrow{\$} |\mathbb{G}|$ is his secret key. Note that y is stored on D's volatile memory. All the data we save will also be stored on D's volatile memory.

3.1 The Main SETUP Attack

In Figure 2 we present the main protocol against UZK. We depict in red the modifications on UZK to obtain our SETUP attack. Note that after session 0 the index is incremented.

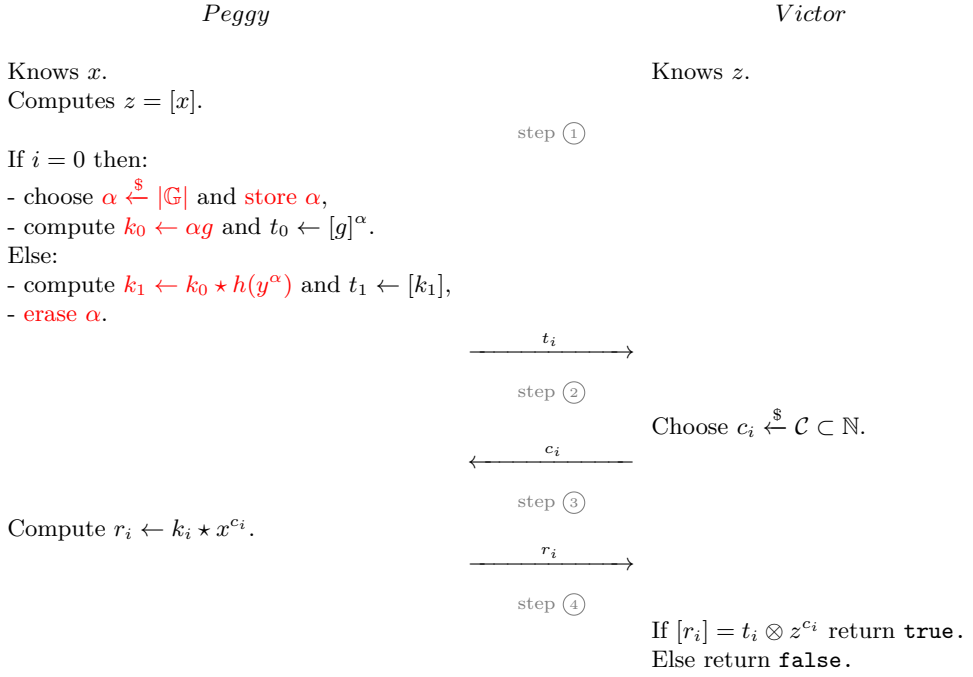

Fig. 2. The Main Unified SETUP Attack.

We further show how *Mallory* can extract *Peggy*'s knowledge if she uses a device that is infected with US-1.

Theorem 2. *If Peggy uses US-1 and UZK satisfies the conditions from Theorem 1, then Mallory can compute an $\tilde{x}$ such that $[\tilde{x}] = z$. More precisely,*

$$\tilde{x} = u^a \star (r_1^{-1} \star r_0 \star h(t_0^{x_M}))^b,$$

where a and b are computed using Euclid's extended gcd algorithm such that $\ell a + (c_0 - c_1)b = 1$.

Proof. From the definitions of r_0 and r_1 we obtain the following relations

$$[r_0] = [k_0 \star x^{c_0}] = t_0 \otimes z^{c_0} \text{ and } [r_1] = [k_1 \star x^{c_1}] = [k_0 \star h(y^{\alpha}) \star x^{c_1}] = t_0 \otimes [h(y^{\alpha})] \otimes z^{c_1}.$$

Let $\beta = h(y^\alpha) = h(t_0^{x_M})$. We make use of

$$[r_1^{-1} \star r_0] = [r_1^{-1}] \otimes [r_0] = z^{-c_1} \otimes [\beta]^{-1} \otimes z^{c_0} = z^{c_0 - c_1} \otimes [\beta]^{-1}$$

and Theorem 1 to see that *Mallory* can compute an $\tilde{x}$ such that $[\tilde{x}] = z$

$$\begin{aligned}
[\tilde{x}] &= [u^a \star (r_1^{-1} \star r_0 \star \beta)^b] \\
&= [u]^a \otimes ([r_1^{-1} \star r_0] \otimes [\beta])^b \\
&= (z^\ell)^a \otimes (z^{c_0 - c_1} \otimes [\beta]^{-1} \otimes [\beta])^b \\
&= z^{\ell a + (c_0 - c_1)b} = z.
\end{aligned}$$

□

We continue by stating the security margin for the IND-SETUP between UZK and US-1.

Theorem 3. *If* HDH *is hard in* $\langle [g] \rangle$ *then UZK and US-1 are* IND-SETUP *in the standard model. Formally, let A be an efficient PPT* IND-SETUP *adversary. There exists an efficient algorithm B such that*

$$ADV^{\textit{IND-SETUP}}_{\textit{UZK, US-1}}(A) \leq 2ADV^{\textit{HDH}}_{\langle [g] \rangle, [g], h}(B).$$

Proof. Let A be an IND-SETUP adversary trying to distinguish between UZK and US-1. We show that A's advantage is negligible. We construct the proof as a sequence of games in which all the required changes are applied to US-1. Let W_i be the event that A wins game i.

Game 0. The first game is identical to the IND-SETUP game[13]. Thus, we have

$$|2Pr[W_0] - 1| = ADV^{\text{IND-SETUP}}_{\text{UZK,US-1}}(A). \tag{1}$$

Game 1. In this game, $h(y^\alpha)$ from *Game 0* becomes $[g]^z$, where $z \xleftarrow{\$} |\mathbb{G}|$. Since this is the only change between *Game 0* and *Game 1*, A will not notice the difference assuming the HDH assumption holds. Formally, this means that there exists an algorithm B such that

$$|Pr[W_0] - Pr[W_1]| = ADV^{\text{HDH}}_{\langle [g] \rangle, [g], h}(B). \tag{2}$$

Game 2. The last change we make is $k_0, k_1 \xleftarrow{\$} \mathbb{G}$. Adversary A will not notice the difference, since

- α is a random exponent and $\mathbb{G}$ is cyclic
- multiplying k_0 with a random element yields a random element.

Formally, we have that

$$Pr[W_1] = Pr[W_2]. \tag{3}$$

The changes made to US-1 in *Game 1* and *Game 2* transformed it into UZK. Thus, we have

$$Pr[W_2] = 1/2. \tag{4}$$

Finally, the statement is proven by combining the equalities (1) − (4). □

[13] as in Definition 5

3.2 A Supplementary SETUP Attack

In Figure 3 we present a supplementary protocol against UZK. Again, we depict in red the modifications made to UZK to obtain our SETUP attack. Note that after session 0 the index is incremented.

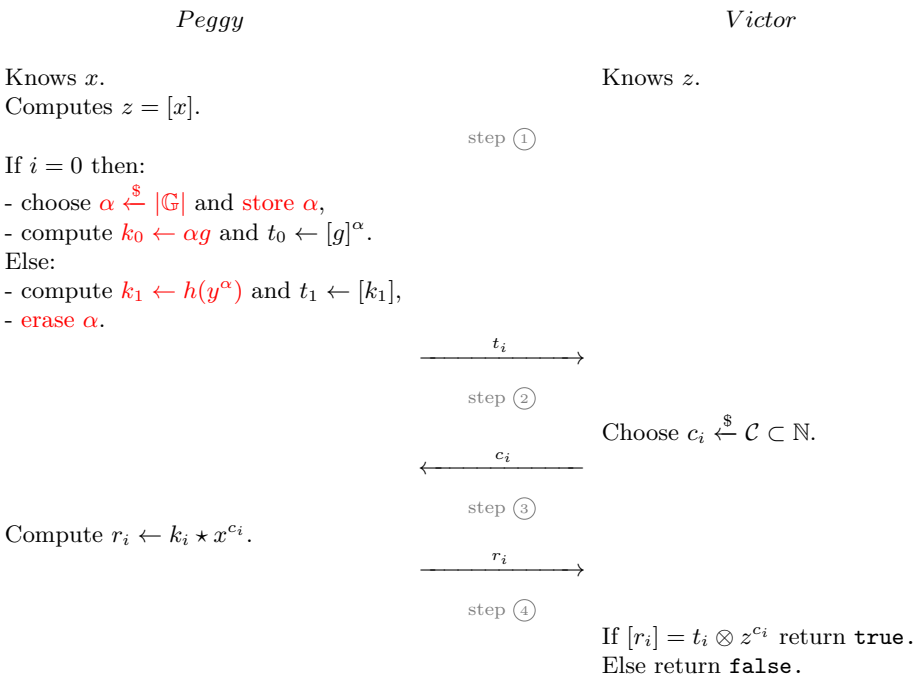

Fig. 3. A Supplementary Unified SETUP Attack.

Unlike US-1, with US-2 *Mallory* cannot extract *Peggy*'s knowledge except for some particular instantiations of UZK. More precisely, if *Mallory* knows or can compute the cardinal of $\mathbb{G}$ then he can extract *Peggy*'s knowledge.

Theorem 4. *If Peggy uses US-2 and $|\mathbb{G}|$ is publicly known, then Mallory can compute an $\tilde{x}$ such that $[\tilde{x}] = z$, with probability $\varphi(|\mathbb{G}|)/|\mathbb{G}|$. More precisely,*

$$\tilde{x} = (r_1 \star (h(t_0^{x_M}))^{-1})^{c_1^{-1}}.$$

Proof. Let $\beta = h(y^{\alpha}) = h(t_0^{x_M})$. From the definition of r_1 we can easily extract x by computing

$$x = (r_1 \star k_1^{-1})^{c_1^{-1}} = (r_1 \star \beta^{-1})^{c_1^{-1}}.$$

□

We further state the security margin for the IND-SETUP between UZK and US-2. We omit the proof due to its similarity to Theorem 3.

Theorem 5. *If HDH is hard in $\langle [g] \rangle$ then UZK and US-2 are IND-SETUP in the standard model. Formally, let A be an efficient PPT IND-SETUP adversary. There exists an efficient algorithm B such that*

$$ADV^{IND\text{-}SETUP}_{UZK,US\text{-}2}(A) \leq 2ADV^{HDH}_{\langle [g] \rangle,[g],h}(B).$$

4 Special Cases of the Unified SETUP Attacks

In this section we describe a number of attacks based on US-1 and US-2 for different instantiations UZK.

4.1 Proofs of Knowledge of a Discrete Logarithm

Let $p = 2q + 1$ be a prime number such that q is also prime. Select an element $h \in \mathbb{H}_p$ of order q in some multiplicative group of order $p - 1$. The discrete logarithm of an element $z \in \mathbb{H}_p$ is an exponent x such that $z = h^x$. We further describe a protocol for proving the knowledge of a discrete logarithm.

The Schnorr protocol [29][14] is a special case of UZK where $(\mathbb{G}, \star) = (\mathbb{Z}_q, +)$ and $\mathbb{H} = \langle h \rangle$. The one-way group homomorphism is defined by $[x] = h^x$ and the challenge space $\mathcal{C}$ can be any arbitrary subset of $[0, q-1]$. According to [22], the conditions of Theorem 1 are satisfied for $\ell = q$ and $u = 0$.

Standard instantiation of the Schnorr protocol define $\mathbb{H}_p$ either as $\mathbb{Z}_p^*$ or as an elliptic curve, so according to Remark 1, we can safely apply both SETUP attacks. Thus, for the first attack we have the following parameters

$$g \leftarrow 1, k_0 \leftarrow \alpha, t_0 \leftarrow h^\alpha, k_1 \leftarrow k_0 + h(y^\alpha), t_1 \leftarrow h^{k_1}.$$

According to Theorem 2, *Peggy*'s secret can be recovered by computing

$$\tilde{x} = (c_0 - c_1)^{-1}(r_0 - r_1 + h(t_0^{x_M})).$$

For the second attack the only change in the protocol is $k_1 \leftarrow h(y^\alpha)$. According to Theorem 4, *Mallory* can recover *Peggy*'s secret by computing

$$\tilde{x} = c_1^{-1}(r_1 - h(t_0^{x_M})).$$

Remark 5. Recovering x when *Peggy* uses US-2 was first described in a series of papers by Young and Yung [32–35]. Remark that in this setting computing x is a little bit more efficient than in the case of US-1.

We further describe a variation of the Schnorr protocol introduced by Girault [18][14]. Thus, let $p = 2fp' + 1$ and $q = 2fq' + 1$ be prime numbers such that f, p' and q' are distinct primes. Select an element $h \in \mathbb{Z}_n^*$ of order f, where $n = pq$. Note that p and q are secret.

Using the UZK notations we have $(\mathbb{G}, \star) = (\mathbb{Z}_f, +)$ and $\mathbb{H} = \langle h \rangle$. The one-way group homomorphism is defined by $[x] = h^x$ and the challenge space $\mathcal{C}$ can be any arbitrary subset of $[0, f - 1]$. It is easy to see that $\ell = f$ and $u = 0$ satisfy the two conditions of Theorem 1.

Since HDH is hard in $\mathbb{H}$[15] then both attacks can be mounted. Note that the attacks can be easily derived from the attacks on the Schnorr protocol.

4.2 Proofs of Knowledge of an e^{th}-root

Let p and q be two safe prime numbers such that $(p - 1)/2$ and $(q - 1)/2$ are also prime. Compute $n = pq$ and choose a prime e such that $\gcd(e, \varphi(n)) = 1$. An e^{th}-root of an element $z \in \mathbb{Z}_n^*$ is a base x such that $z \equiv x^e \bmod n$. Note that the e^{th}-root is not unique. We further describe a protocol for proving the knowledge of an e^{th}-root.

The Guillou-Quisquater protocol [20] is a special case of UZK where $(\mathbb{G}, \star) = (\mathbb{H}, \otimes) = (\mathbb{Z}_n^*, \cdot)$. The one-way group homomorphism is defined by $[x] = x^e \bmod n$ and the challenge space $\mathcal{C}$ can be any arbitrary subset of $[0, e - 1]$. According to [22], the conditions of Theorem 1 are satisfied for $\ell = e$ and $u = z$. Note that when $e = 2$ we obtain the protocol introduced by Fiat and Shamir [15].

[14] This proof can be seen as a more efficient version of a proposal made by Chaum *et al.* [8].

[15] See Remark 1

Remark 6. Before stating the parameters for the SETUP attacks we must first address two issues. The first issue is that both SETUP attacks assume that a generator g is known to *Mallory*. This is needed in order to set-up *Mallory*'s public key. But n is generated internally by *Peggy*'s device and no generator for $\mathbb{Z}_n^*$ is publicly available in the general case. To remove this impediment we always choose $p, q \equiv 3$ or $5 \bmod 8$. According to [23] this ensures us that 2 is a generator for both $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$. Hence, 2 is also a generator for $\mathbb{Z}_n^*$. If p and q are stored only in *Peggy*'s device, then she cannot distinguish this particular choice of primes from other randomly chosen primes, since she only has access to n.

The last issue that we have to address is the selection of *Mallory*'s secret key. Let's assume that n is a λ-bit integer. Since $\phi(n)$ is unknown to *Mallory*, instead of choosing $x_M \xleftarrow{\$} |\mathbb{Z}_n^*|$, he will choose $x_M \xleftarrow{\$} [0, 2^\lambda]$. It is easy to see that the statistical distance between the two distributions is $(\phi(n) - 2^\lambda)/\phi(n)$. Thus, it is negligible.

Since HDH is hard in $\mathbb{H}$[15] and it is infeasible to compute $|\mathbb{G}|$, then only US-1 can be applied. Thus, we have the following parameters for US-1

$$g \leftarrow 2, k_0 \leftarrow 2^\alpha, t_0 \leftarrow 2^{\alpha e}, k_1 \leftarrow k_0 h(y^\alpha), t_1 \leftarrow h^{k_1}.$$

According to Theorem 2, *Peggy*'s secret can be recovered by computing

$$\tilde{x} \equiv z^a \cdot (r_1^{-1} r_0 \cdot h(t_0^{x_M}))^b \bmod n.$$

4.3 Proofs of Knowledge of a Discrete Logarithm Representation

Let $p = 2q + 1$ be a prime number such that q is also prime. Select m elements $h_1, \ldots, h_m \in \mathbb{H}_p$ of order q in some multiplicative group of order $p - 1$. A discrete logarithm representation of an element $z \in \langle h_1, \ldots, h_m \rangle$ is a list of exponents $(x_1, \ldots, x_m)$ such that $z = h_1^{x_1} \ldots h_m^{x_m}$. Note that discrete logarithm representations are not unique. We further describe a protocol for proving the knowledge of a discrete logarithm representation.

A protocol for proving the knowledge of a representation is presented in [22][14]. To instantiate UZK and obtain Maurer's protocol we set $\mathbb{G} = \mathbb{Z}_q^m$ with $\star$ defined as addition applied component-wise and $\mathbb{H} = \langle h_1, \ldots, h_m \rangle$. The one-way group homomorphism is defined by $[(x_1, \ldots, x_m)] = h_1^{x_1} \ldots h_m^{x_m}$ and the challenge space $\mathcal{C}$ can be any arbitrary subset of $[0, q - 1]$. According to [22], the conditions of Theorem 1 are satisfied for $\ell = q$ and $u = (0, \ldots, 0)$. Note that when $m = 2$ we obtain a protocol introduced by Okamoto [25].

The SETUP attacks for this protocol can be easily derived from the attacks on the Schnorr protocol and, thus, are omitted.

Chaum *et al.* [8] also provide a variant for their protocol when n is composite. Thus, by adapting the Girault protocol and tweaking the Maurer protocol, we can obtain a more efficient version of the Chaum *et al.* protocol. Using the notations from the Girault protocol, we set $\mathbb{G} = \mathbb{Z}_f^m$ and $\mathbb{H} = \langle h_1, \ldots, h_m \rangle$, where $h_1, \ldots, h_m \in \mathbb{Z}_n^*$ are elements of order f. The one-way group homomorphism is defined by $[(x_1, \ldots, x_m)] = h_1^{x_1} \ldots h_m^{x_m}$ and the challenge space $\mathcal{C}$ can be any arbitrary subset of $\mathbb{Z}_f$. It is easy to see that $\ell = f$ and $u = (0, \ldots, 0)$. Note that US-1 and US-2 can also be mounted in this setting.

4.4 Proofs of Knowledge of an e^{th}-root Representation

Let p and q be two prime numbers such that $(p - 1)/2$ and $(q - 1)/2$ are also prime. Compute $n = pq$ and choose primes $e_1, \ldots, e_m$ such that $\gcd(e_i, \varphi(n)) = 1$, for $1 \leq i \leq m$. An e^{th}-root representation of an element $z \in \mathbb{Z}_n^*$ is a list of bases $(x_1, \ldots, x_m)$ such that $z \equiv x_1^{e_1} \ldots x_m^{e_m} \bmod n$. Note that e^{th}-root representations are not unique. We further describe a protocol for proving the knowledge of an e^{th}-root representation.

A protocol for proving the knowledge of an e^{th}-root representation can be obtained from UZK if we set $\mathbb{G} = (\mathbb{Z}_n^*)^m$ with $\star$ defined as multiplication applied component-wise and $(\mathbb{H}, \otimes) = (\mathbb{Z}_n^*, \cdot)$. The one-way group homomorphism is defined by $[(x_1, \ldots, x_m)] = x_1^{e_1} \ldots x_m^{e_m} \bmod n$ and the challenge space $\mathcal{C}$ can be any

arbitrary subset of $[0, e-1]$, where e is a prime such that $\gcd(e, \phi(n)) = 1$. Since all e_i are coprime then there exist α_is such that $\alpha_1 e_1 + \ldots + \alpha_m e_m = 1$. Then, it is easy to see that $\ell = 1$ and $u = (z^{\alpha_1}, \ldots, z^{\alpha_m})$.

The US-1 SETUP attack for this protocol can be easily derived from the attack on the Guillou-Quisquater protocol and, thus, is omitted.

5 Conclusions

By introducing a new level of abstraction we devise new attack methods for zero-knowledge protocols and their corresponding signature schemes. It would be interesting to find new protocols that fit our framework.

In [31] we can find an extensive list of signature schemes that are vulnerable to SETUP attacks. Thus, an interesting direction of research is abstracting digital signatures[16] and devising a method for attacking all of them at once, instead of tweaking the attacks for each individual signature.

Acknowledgements

The dissemination of this work is funded by the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 692178.

References

1. Abdalla, M., Bellare, M., Rogaway, P.: DHAES: An Encryption Scheme Based on the Diffie-Hellman Problem. IACR Cryptology ePrint Archive **1999/7** (1999)
2. Ateniese, G., Magri, B., Venturi, D.: Subversion-Resilient Signature Schemes. In: ACM-CCS 2015. pp. 364–375. ACM (2015)
3. Ball, J., Borger, J., Greenwald, G.: Revealed: How US and UK Spy Agencies Defeat Internet Privacy and Security. The Guardian **6** (2013)
4. Bellare, M., Jaeger, J., Kane, D.: Mass-Surveillance without the State: Strongly Undetectable Algorithm-Substitution Attacks. In: ACM-CCS 2015. pp. 1431–1440. ACM (2015)
5. Bellare, M., Paterson, K.G., Rogaway, P.: Security of Symmetric Encryption Against Mass Surveillance. In: CRYPTO 2014. Lecture Notes in Computer Science, vol. 8616, pp. 1–19. Springer (2014)
6. Bellare, M., Rogaway, P.: Minimizing the Use of Random Oracles in Authenticated Encryption Schemes. In: ICICS 1997. Lecture Notes in Computer Science, vol. 1334, pp. 1–16. Springer (1997)
7. Berndt, S., Liśkiewicz, M.: Algorithm Substitution Attacks from a Steganographic Perspective. In: ACM-CCS 2017. pp. 1649–1660. ACM (2017)
8. Chaum, D., Evertse, J.H., Van De Graaf, J.: An Improved Protocol for Demonstrating Possession of Discrete Logarithms and Some Generalizations. In: EUROCRYPT 1987. Lecture Notes in Computer Science, vol. 304, pp. 127–141. Springer (1987)
9. Checkoway, S., Maskiewicz, J., Garman, C., Fried, J., Cohney, S., Green, M., Heninger, N., Weinmann, R.P., Rescorla, E., Shacham, H.: A Systematic Analysis of the Juniper Dual EC Incident. In: ACM-CCS 2016. pp. 468–479. ACM (2016)
10. Checkoway, S., Niederhagen, R., Everspaugh, A., Green, M., Lange, T., Ristenpart, T., Bernstein, D.J., Maskiewicz, J., Shacham, H., Fredrikson, M.: On the Practical Exploitability of Dual EC in TLS Implementations. In: USENIX Security Symposium. pp. 319–335. USENIX Association (2014)
11. Crépeau, C., Slakmon, A.: Simple Backdoors for RSA Key Generation. In: CT-RSA 2003. Lecture Notes in Computer Science, vol. 2612, pp. 403–416. Springer (2003)
12. Dodis, Y., Ganesh, C., Golovnev, A., Juels, A., Ristenpart, T.: A Formal Treatment of Backdoored Pseudorandom Generators. In: EUROCRYPT 2015. Lecture Notes in Computer Science, vol. 9056, pp. 101–126. Springer (2015)
13. Dodis, Y., Gennaro, R., Håstad, J., Krawczyk, H., Rabin, T.: Randomness Extraction and Key Derivation Using the CBC, Cascade and HMAC Modes. In: CRYPTO 2004. Lecture Notes in Computer Science, vol. 3152, pp. 494–510. Springer (2004)
14. Feige, U., Fiat, A., Shamir, A.: Zero-Knowledge Proofs of Identity. Journal of cryptology **1**(2), 77–94 (1988)

[16] not only the ones obtained using the Fiat-Shamir transform

15. Fiat, A., Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: CRYPTO 1986. Lecture Notes in Computer Science, vol. 263, pp. 186–194. Springer (1986)
16. Fried, J., Gaudry, P., Heninger, N., Thomé, E.: A Kilobit Hidden SNFS Discrete Logarithm Computation. In: EUROCRYPT 2017. Lecture Notes in Computer Science, vol. 10210, pp. 202–231. Springer (2017)
17. Gennaro, R., Krawczyk, H., Rabin, T.: Secure Hashed Diffie-Hellman over Non-DDH Groups. In: EUROCRYPT 2004. Lecture Notes in Computer Science, vol. 3027, pp. 361–381. Springer (2004)
18. Girault, M.: An Identity-based Identification Scheme Based on Discrete Logarithms Modulo a Composite Number. In: EUROCRYPT 1990. Lecture Notes in Computer Science, vol. 473, pp. 481–486. Springer (1990)
19. Gordon, D.: Designing and Detecting Trapdoors for Discrete Log Cryptosystems. In: CRYPTO 1992. Lecture Notes in Computer Science, vol. 740, pp. 66–75. Springer (1993)
20. Guillou, L.C., Quisquater, J.J.: A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Transmission and Memory. In: EUROCRYP 1988. Lecture Notes in Computer Science, vol. 330, pp. 123–128. Springer (1988)
21. Maimuţ, D., Teşeleanu, G.: Secretly Embedding Trapdoors into Contract Signing Protocols. In: SECITC 2017. Lecture Notes in Computer Science, vol. 10543. Springer (2017)
22. Maurer, U.: Unifying Zero-Knowledge Proofs of Knowledge. In: AFRICACRYPT 2009. Lecture Notes in Computer Science, vol. 5580, pp. 272–286. Springer (2009)
23. McCurley, K.: A Key distribution System Equivalent to Factoring. Journal of cryptology **1**(2), 95–105 (1988)
24. Naor, M., Reingold, O.: Number-Theoretic Constructions of Efficient Pseudo-Random Functions. Journal of the ACM (JACM) **51**(2), 231–262 (2004)
25. Okamoto, T.: Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In: CRYPTO 1992. Lecture Notes in Computer Science, vol. 740, pp. 31–53. Springer (1992)
26. Perlroth, N., Larson, J., Shane, S.: NSA Able to Foil Basic Safeguards of Privacy on Web. The New York Times **5** (2013)
27. Russell, A., Tang, Q., Yung, M., Zhou, H.S.: Cliptography: Clipping the power of kleptographic attacks. In: ASIACRYPT 2016. Lecture Notes in Computer Science, vol. 10032, pp. 34–64. Springer (2016)
28. Russell, A., Tang, Q., Yung, M., Zhou, H.S.: Generic Semantic Security against a Kleptographic Adversary. In: ACM-CCS 2017. pp. 907–922. ACM (2017)
29. Schnorr, C.P.: Efficient Identification and Signatures For Smart Cards. In: CRYPTO 1989. Lecture Notes in Computer Science, vol. 435, pp. 239–252. Springer (1989)
30. Shoup, V.: Sequences of Games: A Tool for Taming Complexity in Security Proofs. IACR Cryptology ePrint Archive **2004/332** (2004)
31. Teşeleanu, G.: Threshold Kleptographic Attacks on Discrete Logarithm Based Signatures. IACR Cryptology ePrint Archive **2017/953** (2017)
32. Young, A., Yung, M.: The Dark Side of Black-Box Cryptography or: Should We Trust Capstone? In: CRYPTO 1996. Lecture Notes in Computer Science, vol. 1109, pp. 89–103. Springer (1996)
33. Young, A., Yung, M.: Kleptography: Using Cryptography Against Cryptography. In: EUROCRYPT 1997. Lecture Notes in Computer Science, vol. 1233, pp. 62–74. Springer (1997)
34. Young, A., Yung, M.: The Prevalence of Kleptographic Attacks on Discrete-Log Based Cryptosystems. In: CRYPTO 1997. Lecture Notes in Computer Science, vol. 1294, pp. 264–276. Springer (1997)
35. Young, A., Yung, M.: Malicious Cryptography: Exposing Cryptovirology. John Wiley & Sons (2004)
36. Young, A., Yung, M.: Malicious Cryptography: Kleptographic Aspects. In: CT-RSA 2005, Lecture Notes in Computer Science, vol. 3376, pp. 7–18. Springer (2005)

A Additional Preliminaries

Definition 6 (Computational Diffie-Hellman - CDH). *Let $\mathbb{D}$ be a cyclic group of order q, d a generator of $\mathbb{D}$ and let A be a probabilistic polynomial-time algorithm (PPT algorithm) that returns an element from $\mathbb{D}$. We define the advantage*

$$ADV^{CDH}_{\mathbb{D},d}(A) = Pr[A(d^x, d^y) = d^{xy} | x, y \xleftarrow{\$} \mathbb{Z}_q^*].$$

If $ADV^{CDH}_{\mathbb{D},d}(A)$ is negligible for any PPT algorithm A, we say that the Computational Diffie-Hellman problem is hard in $\mathbb{D}$.

Definition 7 (Decisional Diffie-Hellman - DDH). *Let $\mathbb{D}$ be a cyclic group of order q, g a generator of $\mathbb{D}$. Let A be a PPT algorithm which returns 1 on input (d^x, d^y, d^z) if $d^{xy} = d^z$. We define the advantage*

$$ADV^{DDH}_{\mathbb{D},d}(A) = |Pr[A(d^x, d^y, d^z) = 1 | x, y \xleftarrow{\$} \mathbb{Z}^*_q, z \leftarrow xy] - Pr[A(d^x, d^y, d^z) = 1 | x, y, z \xleftarrow{\$} \mathbb{Z}^*_q]|.$$

If $ADV^{DDH}_{\mathbb{D},d}(A)$ is negligible for any PPT algorithm A, we say that the Decisional Diffie-Hellman problem is hard in $\mathbb{D}$.

Definition 8 (Entropy Smoothing - ES). *Let $\mathbb{D}$ be a cyclic group of order q, $\mathcal{K}$ the key space and $\mathcal{H} = \{h_i\}_{i \in \mathcal{K}}$ a family of keyed hash functions, where each h_i maps $\mathbb{D}$ to $\mathbb{E}$, where $\mathbb{E}$ is a group. Let A be a PPT algorithm which returns 1 on input (i, y) if $y = h_i(z)$, where z is chosen at random from $\mathbb{D}$. Also, let We define the advantage*

$$ADV^{ES}_{\mathcal{H}}(A) = |Pr[A(i, h_i(z)) = 1 | i \xleftarrow{\$} \mathcal{K}, z \xleftarrow{\$} \mathbb{D}] - Pr[A(i, h) = 1 | i \xleftarrow{\$} \mathcal{K}, h \xleftarrow{\$} \mathbb{E}]|.$$

If $ADV^{ES}_{\mathcal{H}}(A)$ is negligible for any PPT algorithm A, we say that $\mathcal{H}$ is Entropy Smoothing.

Remark 7. In [13], the authors prove that the CBC-MAC, HMAC and Merkle-Damgård constructions satisfy the above definition, as long as the underlying primitives satisfy some security properties.