Applied Mathematics and Computation 333 (2018) 467–479

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Still wrong use of pairings in cryptography

Osmanbey Uzunkol[a,b,*], Mehmet Sabır Kiraz[b]

[a] FernUniversität in Hagen, Fakulty of Mathematics and Computer Science, Universitätsstr. 1 (IZ), D-58097 Hagen, Germany
[b] Mathematical and Computational Sciences Labs, TÜBİTAK BİLGEM, P.O. BOX: 74, 41470, Gebze/Kocaeli, Turkey

ARTICLE INFO

MSC:
94A60

Keywords:
Pairing-based protocols
Bilinear maps
Security
Efficiency
The discrete logarithm problem

ABSTRACT

Recently many pairing-based cryptographic protocols have been designed with a wide variety of new novel applications including the ones in the emerging technologies like cloud computing, internet of things (IoT), e-health systems, and wearable technologies. There have been, however, a wide range of incorrect use of these primitives mainly because of their use in a “black-box” manner. Some new attacks on the discrete logarithm problem lead to either totally insecure or highly inefficient pairing-based protocols, and extend considerably the issues related to pairings originally pointed out by Galbraith et al. (2008). Other reasons are the implementation attacks, the minimal embedding field attacks, and the issues due to the existence of auxiliary inputs. Although almost all these issues are well-known to mathematical cryptographers, there is no state-of-the-art assessment covering all these new issues which could be used by the applied cryptography researchers and the IT-security developers. In order to illustrate this point, we give a list of recent papers having either wrong security assumptions or realizability/efficiency issues. Furthermore, we give a compact and an state-of-the-art recipe of the correct use of pairings for the correct design with a view towards efficient and secure implementation of security solutions using these primitives.

1. Introduction

Pairing-based cryptography has received much attention because of wide variety of its immediately deployable applications. These applications include identity-based encryption, functional and attribute-based encryption, searchable encryption, short/group/ring signatures, signcryption, homomorphic linear authenticators for integrity checking, security, privacy and integrity solutions for cloud computing and Internet of Things (IoT), e-health systems, and wearable technologies. We refer to Appendix for a selected list of some novel applications using pairing-based cryptography. In practice, Voltage Security (now an HP company) and Trend Micro are the most well-known companies utilizing the pairing-based security solutions [66].

There have been unfortunately a collection of recent results using the pairings incorrectly due to not being aware of the recent advancements on solving the discrete logarithm problems in some groups. We observed that there are unfortunately plenty of very recently introduced papers (surprisingly) either having pairing related wrong security assumptions and/or efficiency issues.

[*] Corresponding author at: FernUniversität in Hagen, Fakulty of Mathematics and Computer Science, Universitätsstr. 1 (IZ), D-58097 Hagen, Germany.
E-mail addresses: osmanbey.uzunkol@gmail.com, osmanbey.uzunkol@fernuni-hagen.de (O. Uzunkol), mehmet.kiraz@tubitak.gov.tr (M.S. Kiraz).

https://doi.org/10.1016/j.amc.2018.03.062

The security of pairing-based cryptosystems relies on the difficulty of various computationally hard problems related to the discrete logarithm problem (DLP). The new attacks on the DLP on some groups [3,9,37,39,69] have significant consequences on the security of some pairings primitives. Furthermore, very recent results on solving the DLP for finite fields of medium characteristics and composite degrees size have also consequences on the choice of key sizes for pairing based cryptography [8,45,48,72]. Hence, ignoring these recent technical advancements in solving the DLP make certain security assumptions incorrect. We note that although some basic problems related to using pairings as "black boxes" incorrectly was introduced by Galbraith et al. [35], not being aware of of these new issues is the primary reason of designing protocols which have considerably critical security vulnerabilities, realizability issues and/or efficiency problems. The complexity of these mathematical preliminaries is undoubtedly the reason of neglecting the realization concerns in the design of pairing-based protocols.

1.1. Our contribution

The main contributions can be listed as follows:

- Firstly, we highlight the main issues related to the correct use of pairings by revisiting the most recent attacks against pairing-based cryptography. These include new advancements in the discrete logarithm problem, implementation attacks, and others (e.g., protocols based on the discrete logarithm problem with auxiliary inputs, minimum embedding degree attacks).
- We give secondly a new assessment of the correct use of pairings with an informative and less technical way (as a recipe for designers and developers) extending considerably the issues already introduced by Galbraith et al. [35]. Thereby, we show the serious effects of the recent attacks on the designs, security models, and hardness assumptions of cryptographic protocols related to efficient and secure realization of pairing-based real-world applications.
- Following the lines of our assessment, we present security and/or efficiency issues of many recent papers. In particular, some of these issues could easily be solved by using smaller key but larger ciphertext sizes whereas the others unfortunately nullify the contribution because of unrealizable and/or insecure use of pairings.

2. Basics for pairing-based cryptography

We begin with the abstract pairing requirements and different types of bilinear maps used in cryptographic protocols.

Let $(\mathbf{G}_1, +)$ and $(\mathbf{G}_2, +)$ be two additive cyclic groups of (nearly) prime order q with $\mathbf{G}_1 =< P >$ and $\mathbf{G}_2 =< Q >$, $(\mathbf{G}_T, \cdot)$ be a multiplicative cyclic group of order q with $\mathbf{G}_T =< g >$. We write as usual 0 for the identity elements of $\mathbf{G}_1$, $\mathbf{G}_2$ and 1 for $\mathbf{G}_T$. A *pairing* or a *bilinear map* is a map $e: \mathbf{G}_1 \times \mathbf{G}_2 \rightarrow \mathbf{G}_T$ satisfying the following properties:

- *Bilinearity:* For all $P_1, P_1' \in \mathbf{G}_1, Q_1, Q_1' \in \mathbf{G}_2$, e is a group homomorphism in each component, i.e.
  1. $e(P_1 + P_1', Q_1) = e(P_1, Q_1) \cdot e(P_1', Q_1)$,
  2. $e(P_1, Q_1 + Q_1') = e(P_1, Q_1) \cdot e(P_1, Q_1')$.
- *Non-degeneracy:* e is non-degenerate in each component, i.e.
  1. For all $P_1 \in \mathbf{G}_1$, $P_1 \neq 0$, there is an element $Q_1 \in \mathbf{G}_2$ such that $e(P_1, Q_1) \neq 1$,
  2. For all $Q_1 \in \mathbf{G}_2$, $Q_1 \neq 0$, there is an element $P_1 \in \mathbf{G}_1$ such that $e(P_1, Q_1) \neq 1$.
- *Computability:* There exists an algorithm which computes the bilinear map e efficiently.

There are essentially 4 types of bilinear maps [35,74] used in the design of pairing-based protocols depending on the special requirements such as short representation, hashing to a group element, efficient homomorphisms.

- *Type-1:* $\mathbf{G}_1 = \mathbf{G}_2$. In this case there exists no short representations for the elements of $\mathbf{G}_1$.
- *Type-2:* $\mathbf{G}_1 \neq \mathbf{G}_2$ and there is an efficiently computable homomorphism ϕ: $\mathbf{G}_2 \rightarrow \mathbf{G}_1$. In this case no efficient secure hashing to the elements in $\mathbf{G}_2$ is possible.
- *Type-3:* $\mathbf{G}_1 \neq \mathbf{G}_2$ and there exists no efficiently computable homomorphism ϕ: $\mathbf{G}_2 \rightarrow \mathbf{G}_1$.
- *Type-4:* $\mathbf{G}_1 \neq \mathbf{G}_2$ and there exists an efficiently computable homomorphism ϕ: $\mathbf{G}_2 \rightarrow \mathbf{G}_1$ as in the case of the Type-2 setting but with an efficient secure hashing method to a group element [74]. Security proofs can be quite cumbersome in this setting as discussed in [50]. We note that this type is not generally used in protocol designs due to its inefficiency.

The main disadvantage of the Type-2 pairing is that there exists no random sampling algorithm from $\mathbf{G}_2$ (yielding to a secure hash function) which maps arbitrary elements to $\mathbf{G}_2$, [35, pp. 3119]. Note that there exists a natural, efficient, and secure transformation of protocols using the Type-2 pairing into protocols using the Type-3 pairing [18, Section 5]. We summarize this fact as follows since we need this in the subsequent sections:

Fact 1. For Type-2 pairings there exists no random sampling algorithm from $\mathbf{G}_2$ mapping arbitrary elements into $\mathbf{G}_2$. Therefore, Type-2 pairings cannot have secure hash functions into the group $\mathbf{G}_2$.

Remark 1. We note that most pairing-based aggregate signature protocols require an efficiently computable homomorphism ϕ: $\mathbf{G}_2 \rightarrow \mathbf{G}_1$, i.e., they use pairings of Type-2, see for instance [15].

The Type-1 setting is commonly called *symmetric pairing* while other types are called *asymmetric pairing*.