Distributed Cryptanalytic Time-Memory Trade-Offs

Contact: Gildas AVOINE (gildas.avoine@irisa.fr)
IRISA, INSA Rennes, France

Keywords
Cybersecurity, Cryptography, Distributed Computing, Password Cracking, Privacy.

Synopsis
Information security is usually based on secrets known by authorized parties only, e.g., passwords, PIN codes, and cryptographic keys. Breaking a system means either circumventing the security mechanisms, or exploiting a weakness in these mechanisms, or retrieving the secrets by any other means, including through an exhaustive search among all possible secrets. If secrets were long-enough and truly random, the latter attack would not be practical. It is unfortunately very common that real-life secrets do not fulfill these assumptions, meaning that they do not carry enough entropy.

A cryptanalytic time-memory trade-off (TMTO) is a technique introduced by Martin Hellman in 1980 to speed up exhaustive searches using stored precomputations. TMTOs were not efficient in the eighties due to the limited resources of that time’s computers, and they have no longer been considered until Philippe Oechslin resurrected them in 2003 to crack Windows passwords. His contribution was outstanding: he demonstrated his ability to crack any alphanumerical Windows password within a few seconds. Today, TMTOs belong to any security expert’s toolbox, e.g., to perform penetration tests, legal investigations, and to test users’ passwords on the fly when they are requested to refresh their passwords. TMTOs’ impact is not limited to passwords, though, and they have been successfully used to attack for example GSM’s security primitives and vehicle ignition systems.

This thesis aims at improving TMTOs along two axes: improving the precomputation phase using a distributed approach, and applying the technique to the field of data desanonymisation.

Indeed, the main bottleneck of TMTOs is today the precomputation phase, which requires large resources, while most of the precomputations are actually wasted (typically, 96% of the precomputations are eventually thrown away). This offers room for progress. Although there exits a method to make the precomputation phase more efficient, it cannot be used when considering a distributed precomputation phase, which is yet an unavoidable setup for large-scale TMTOs. We already performed promising experiments on a new way to perform a distributed precomputation phase, and the thesis will formalize and optimize this approach, and discover new approaches.

Then, we would like to highlight that anonymisation of data is commonly achieved by applying a one-way function on the field that identifies the data. It can for example be names, email addresses, mac addresses, etc. Desanonymizing data is then very close from cracking passwords. However, no one attempted yet to apply the TMTO technique for desanonymizing data. We expect to do so in this thesis, which may lead to significant scientific advances and a high societal impact.
**Context**

Many cryptanalytic problems can be solved using an exhaustive search in the key space, but each new instance of the problem requires restarting this expensive process from scratch. The basic idea of a cryptanalytic time-memory trade-off (TMTO) is to carry out an exhaustive search once for all such that following instances of the problem become easier to solve. Thus, if there are $N$ possible solutions to a given problem, and $M$ units of memory are allocated for the attack, then a time-memory trade-off can solve the problem with $T$ units of time where $T$ is proportional to $N^2/M^2$, instead of $T = N$ with an exhaustive search [2, 4]. TMTOs are used to perform chosen plaintext attacks when $N$ is reasonably sized. TMTOs are for example available in any security expert's toolbox to crack passwords [5], especially when the attack must be performed in a short time, what is called a lunch-time attack.

**Background**

The cryptanalytic time-memory trade-off has been introduced in 1980 by Hellman [3] and applied to DES. Given a plaintext $P$ and a ciphertext $C$, the problem consists in recovering the key $K$ such that $C = E_K(P)$, where $E$ is an encryption function assumed to follow the behavior of a random function. Encrypting $P$ under all possible keys and storing each corresponding ciphertext allows for immediate cryptanalysis but needs $N$ elements of memory. The idea of a trade-off is to use chains of keys, which is done using a reduction function $R$ that generates a key from a ciphertext. Using $E$ and $R$, chains of alternating ciphertexts and keys can thus be generated. The key point is that only the first and the last element of each chain are stored. This process is called the precomputation phase and it is done well in advance before the attack. Then, during the attack itself, in order to retrieve $K$, a chain is generated from $C$. If at some point it yields a stored end of chain, then the entire chain is regenerated from its starting point. However, finding a matching end of chain does not necessarily imply that the key will be found in the regenerated chain. There exist situations where the chain that has been generated from $C$ merges with a chain that is stored in the memory that does not contain $K$. This situation is called a false alarm and is very costly.

**Research Issue**

Currently, it is hard to perform a TMTO when $N$ is large, let’s say when $N > 2^{48}$ because the precomputation phase is quite long. Unfortunately, distributing the precomputation phase over several computers, e.g., on a 5'000-core cluster, is not as efficient as one may expect. For example, while someone may
expect a 5’000 speed-up factor with the above-mentioned cluster, it will not be higher than 100 with
the current techniques used for the precomputations. The reason is that all the processes must exchange
messages all along the precomputation phase in order to discard duplicated values as early as possible.
The objective of this thesis is to find a solution to make a distributed precomputation phase as efficient
as possible, given the optimal bound is reached when the communication overhead is null. Note that this
case never occurs because there is a communication overhead even when considering a single processing
unit because the computed values must be temporarily stored in a cache memory. In this thesis, it will
be important to consider both the computation and the communication parameters in order to suit the
algorithm used for the precomputation phase to the considered processing units (CPU, GPU, FPGA) and
memories (RAM, SSD, etc.).

The second axis of the thesis consists in exploring the feasibility to use TMTOs to desanonymize
data. It is indeed common in practice to get access to open data whose anonymisation simply consists
in applying a hash function on the data’s key field. For example, when considering logs on a local area
network, the key field of the data can be the mac addresses of the computers on the network. A TMTO
can then be applied because the set of MAC addresses is small enough in practice to apply a TMTO.
In particular, MAC addresses are not uniformly distributed (they are strongly related to the network
interface’s manufacturer), and we already demonstrated[1] that TMTOs are much more efficient when
the values to be recovered are not uniformly distributed.

References
trade-offs on non-uniform distributions. In Computer Security - ESORICS 2015 - 20th European
Symposium on Research in Computer Security, Vienna, Austria, September 21-25, 2015, Proceed-
using checkpoints. In Progress in Cryptology – Indocrypt 2005, volume 3797 of Lecture Notes in
of India, Springer-Verlag.
Advances in Cryptology – CRYPTO’03, volume 2729 of Lecture Notes in Computer Science, pages
617–630, Santa Barbura, California, USA, August 2003. IACR, Springer-Verlag.

Research Institute
IRISA (Institut de Recherche en Informatique et Systèmes Aléatoires), founded in 1975, is a research
center for IT, image, signal processing, and robotics, located in Rennes, France. The institute hosts 800
researchers distributed in 40 research groups, and is funded by 8 entities, namely CNRS, ENS Rennes,
Inria, INSA Rennes, Institut-Mines-Télécom, CentraleSupélec, Université de Bretagne Sud (UBS), and
Université de Rennes 1. IRISA so “forms a research cluster for excellence within the ICTS, with scien-
tific priorities that include bioinformatics, system security, new software architecture (many-cores, cloud
computing), and virtual reality”. IRISA is well-known for its research activities in computer security and
cryptography (more than 120 researchers work full-time on this topic) and many neighboring companies
are actively involved in this field. Rennes is located in the West part of France, about 45 minutes by car
from the sea, and a fast train connects Rennes to Paris in less than 1h30.
**Research Group**

Embedded Security and Cryptography (EMSEC) is a research group within the IRISA computer science institute located in Rennes, France. EMSEC was created in February 2016 and is headed by Prof. Gildas Avoine and Prof. Pierre-Alain Fouque. The group hosts more than 35 researchers, including 8 permanent members. EMSEC’s activities are organized along three axes: cryptography, formal methods, and system security.

Link: [https://www.irisa.fr/emsec](https://www.irisa.fr/emsec)