OMEN：基于马尔科夫模型可能性递减序列的更快速的密码猜测方法

Markus Dürmuth[1], Fabian Angelstorf[1], Claude Castelluccia[2], Daniele Perito[2], and Abdelberi Chaabane[2]

[1] Ruhr-University Bochum, Germany

markus.duermuth@rub.de

[2] INRIA, France

{claude.castelluccia,daniele.perito}@inria.fr

摘要

密码被广泛运用在用户验证，尽管现在一些密码安全性较差，但是它还将在可预见的未来中长久的发挥功能。密码安全性较差的重要原因是用户创造密码的时候太有规律可循，使得太容易受到密码猜测攻击。理解攻击者的密码猜测攻击的方法是非常重要，也是非常必需的，这样我们才能更好的寻找对策来抑制他们的攻击。

这篇文章使用流行的 OMEN 方法，即被 Narayanan 和 Shmatikov（CCS 2005）提出的一种新的基于马尔科夫模型的密码破解方法。主要的创新点是我们的工具在生成密码时是根据它们最近的出现概率，并且首先生成的密码更像是“密码”。我们大量的实验表明， 相比较与现有的一些方法，OMEN 显著的提升了密码猜测速度。

我们还针对 Narayanan 和 Shmatikov 使用的 John the Ripper 的马尔科夫模型做了性能对比实验。OMEN 猜测方法在第一组 9000 万次猜测中的正确率提高了 40%，而 John the Ripper 的马尔科夫模型需要至少 8 倍的猜测量才能达到同等的正确率。并且 OMEN 猜测方法在 100 亿次猜测中超过了我们比较的所有的密码猜测方法。

关键词：验证，密码猜测，马尔科夫方法

1 引言

无论是在线还是离线，基于密码的用户验证都是最广泛的运用于用户验证的方法。尽管有一些密码安全性较差，在可预见的未来，密码还将会是用户验证的最主要的一种形式，因为大量的优点：密码有高可移植性，易于被用户理解，易于被开发者管理。事实上，可选择性的用户验证形式有在特殊的场景当中替代密码的能力，虽然现在他们还不行，但是将来能在大范围的场景当中替代密码。

在这次工作中，我们专注于离线的密码猜测攻击，攻击者可以进行大量的密码猜测，仅仅受限于工作者愿意投入的时间和资源。这种攻击是可以通过攻击者提高生成和验证猜测的计算资源来大幅度提高速度的，比如使用特定的硬件设施和大量的计算资源等。我们专注于通过技术来减少破解密码需要的猜测次数。因此，我们的目标是在不依赖于充足的资源的情况下减少攻击次数。

类似于字典模型中的 John the Ripper (JtR)（对现存的单词字段使用 mangling rules 来挖掘密码的结构规律)这类工具是非常常用的，这被用来通过现有的字典或者最近泄漏的密码数据库之类的数据全集来生成新的密码。Weir et al. 示范了如何使用上下文无关文法（PCFG）来从泄漏的密码数据库中自动获取 mangling 规则，Narayanan et al 展示了因为与自然语言处理紧紧相关而闻名的马尔科夫模型，也能被用来高效的猜测密码。我们将证明虽然这些攻击方法已经能获得一个高效的密码破解，但是他们的性能还是能有实质性的提升。

这篇文章将介绍 OMEN, 一种新的基于马尔科夫模型的攻击，通过最近出现的可能性来生成密码，生成的密码更接近真正的“密码”。通过我们大量的实验，比起现存的方法，OMEN 显著的提高了猜测速度。

1.1 相关工作

密码存在一个很大的问题是用户可能选择一个安全性很弱的密码。这些密码普遍有极强的结构性，所以可以被粗暴的密码猜测攻击更快速的攻破。为了保护在数据库被攻破时泄漏的纯文本信息，最好的授权实现是仅把 hash 以后的密码保存在服务器中，而不是明文密码。

在这次工作当中，我们考虑离线的猜测攻击，并且假设攻击者已经获得了 hash 方法并且恢复了密码。hash 函数通常被设计来降低攻击者猜测的企图，这意味着泄漏的风险主要受 hash 函数的计算代价主导。所以我们为了用更小的代价来猜测一个新的密码，我们在评估所有的攻击者的时候我们基于它进行猜测直到猜测正确的次数。

John the Ripper：John the Ripper 是一个非常著名的密码攻击者。它经常提出不同的方法来生成密码。在字典模型中，一个单词字典作为提供的输入，这个工具可以对其中的所有单词进行测试。用户能特定多种不同的 mangling 规则。就像[6]，我们发现小部分相关联的猜测（少于10^8），在字典模型当中，JtR 是最好的规则，在加强的模型（JtR-inc）中，JtR 尝试通过 3 阶算子的马尔科夫模型来猜测密码。

Password Guessing with Markov Models：马尔科夫模型已经证明其在计算机安全领域，

特别是密码安全领域非常的有用。我们有一个很有用的工具去破解密码，并且我们同样能用它来正确的估计新密码的安全强度。最近一个独立的工作比较了不同的几种概率密码模型，并且得出结论，相比上下文无关文法模型，马尔科夫模型是更适合来估计密码概率。我们的工作和他们相比最大的不同是，他们仅使用类似的密码，没有制作出一个产出正确密码顺序的密码攻击者。但这是我们工作的主要贡献。

马尔科夫模型的基本思想是密码中的毗邻的字母不是独立的选择，而是遵循一种确定的规律（比如 2 阶的 th 比起 tq 是有更高的可能性的，e 有很高的可能性出现在 th 后面）。在 n 阶的马尔科夫模型当中，字符串中下一个字符出现的概率基于以前的 n-1 个字符。因此，对于给定的字符串 $c_1, \dots, c_m$，马尔科夫模型估算的概率是$P(c_1, \dots, c_m) \approx P(c_1, \dots, c_{n-1}) \cdot \prod_{i=n}^{m} P(c_i | c_{i-n+1}, \dots, c_{i-1})$。对于密码攻击，需要通过真实的数据集知道初始概率$P(c_1, \dots, c_{n-1})$和转移概率$P(c_n | c_1, \dots, c_{n-1})$（它应该尽可能的接近我们攻击的数据的分布），然后按马尔科夫模型估计的可能性递减顺序枚举密码。为了使得攻击高效，我们需要考虑一些细节，有限的数据是一个挑战（数据稀少），最优的密码枚举顺序是一个挑战。

Probabilistic Grammars-Based Schemes：一个基于概率的上下文无关文法，该方法首先假设典型的密码有确定的结构。通过一系列真实的密码数据来获得不同结构的可能性，然后使用这些结构来生成密码。

Password Strength Estimation：有一个问题与密码猜测非常相关，就是评估密码的强度。知道密码准确的安全等级对系统管理者而言是非常重要、关键的。在最开始的时候，密码攻击者习惯于寻找安全性弱的密码，现在很多改进的版本被开发了出来。典型的是，一些专业的活跃的密码破解手段被用来排除安全性弱的密码。但是，很多活跃的密码破解手段几乎都是使用简单的规则集合去确定密码的强度，这些方法在真实的密码中表现出了较差的强度估计能力。[10] 研究了密码政策在密码强度上的影响性，[2] 提出了一种新的测量密码强度并且应用于大型的数据集和密码中。最近，Schechter et al.通过限制一个密码的次数来对密码强度进行分类。最后，根据可靠的证实，马尔科夫模型应该显示出了对密码强度很好的预言能力。

1.2 文章组织

在第 2 部分，我们描述了马尔科夫模型可能性递减序列方法（OMEN） 和为了选择满意的参数而设计的几组实验。在第 3 部分，给了一些有关 OMEN 攻击性能的细节，包括与别的密码猜测方法进行比较。我们在第 4 部分对文章进行了简介的总结。

2 OMEN：一个加强的马尔科夫模型密码破解方法

在这一部分，我们介绍基于马尔科夫模型的密码列举算法的实现，enumPwd()。我们的实现提高了之前 Narayanan et al.[16]和 JtR[17]的马尔科夫模型的方法。然后我们介绍 OMEN 是如何工作的，实践我们的新的密码破解器。

2.1 一种加强的列举算法 enumPwd()

Narayanan et al. 索引算法有一个缺点，不能按概率递减顺序输出密码。但是，以正确的顺序进行密码猜测能够显著的提高密码猜测速度（可以看我们第 3 部分的例子）。我们开发了一种算法 enumPwd()，大致按照概率递减的顺序来列举密码。

在一个很高的程度上，我们的算法离散了所有的可能性到一系列箱子中，然后迭代所有的箱子以达到概率递减的顺序。对于每一个箱子，寻找所有能够与其相匹配概率的密码，然后输出它们。更精确的是，我们先计算 n 阶概率的对数，然后像 Narayanan et al.［16］一样离散它们到多个等级。根据公式 $lvl_i = round(\log(c_1 \cdot prob_i + c2))$ ，这里c_1和c_2是被选择来让最高频率的 n 阶字符串是第 0 等级，不会出现在训练集中的 n 阶字符串依然会被分配一个很小的概率。可以看到这里等级是负数，我们调整参数去获得一些理想的等级(nbLevel)，比如，等级可以是 $0, -1 \ldots, -(\text{nbLevel}-1)$， 这里的 nbLevel 是一个参数。等级的数值影响着算法的准确度和运行时间：更高的等级以为着更好的准确度和更长的运行时间。

对于一个具体的长度φ和等级η， $enumPwd(\eta, \varphi)$ 依照以下的步骤执行：

1. 计算长度为 $\varphi - 1$ 的所有向量 $\mathrm{a} = (\mathrm{a}_2, \ldots, \mathrm{a}_\varphi)$，每一个元素 a_i 是一个在［0, nbLevel − 1］范围内的整数，并且所有元素的和为 η 。所有的向量有 $\varphi - 1$ 个元素，当我们使用三阶算子时，我们需要 $\varphi - 2$ 个转移概率和 1 个初始概率去确定长度为 φ 的字符串的可能性。比如，对于长度为 $\varphi = 8$ 的密码“password”的概率按照以下的公式计算：

$$\mathrm{P}(\text{password}) = P(pa)P(s|pa)P(w|ss)P(o|sw)P(r|wo)P(d|or)$$

2. 对于每一个向量 a ，选择所有概率等于等级 a_2 的二阶算子 x_1x_2 ，对于每一个二阶算子，循环的匹配 x_3 组成概率等于等级 a_3 的三阶算子 $x_1x_2x_3$ 。接下来，对于每一个三阶算子，循环的匹配 x_4 组成概率等于等级 a_4 的三阶算子 $x_2x_3x_4$ 。依次类推，直到达到想要的长度。最后，输出一个长度为 φ，等级（或者“强度”）为 η 的候选密码集合。

Algorithm 1 Enumerating passwords for level η and length ℓ (here for $\ell = 4$).[2]

function enumPwd(η, ℓ)

1. for each vector $(a_i)_{2 \le i \le \ell}$ with $\sum_i a_i = \eta$
   and for each $x_1x_2 \in \Sigma^2$ with $L(x_1x_2) = a_2$
   and for each $x_3 \in \Sigma$ with $L(x_3 \mid x_1x_2) = a_3$
   and for each $x_4 \in \Sigma$ with $L(x_4 \mid x_2x_3) = a_4$:
   (a) output $x_1x_2x_3x_4$

算法 1 是一个更加公式化的描述。它描述了当 $\varphi = 4$的时候的算法。但是，拓展到更大的 φ 也是非常直观的。

例子：我们用一个直观的例子来描述算法。为了简单一点，我们考虑密码的长度 $\varphi = 3$ ，字母表 $\sum = \{a, b\}$ ，初始概率等级为：

$$\mathrm{L(aa)} = 0, L(ab) = -1$$
$$\mathrm{L(ba)} = -1, L(bb) = 0$$

转移概率等级为：

$$\mathrm{L}(a|aa) = -1, L(b|aa) = -1$$
$$\mathrm{L}(a|ab) = 0, L(b|ab) = -2$$
$$\mathrm{L}(a|ba) = -1, L(b|ba) = -1$$
$$\mathrm{L}(a|bb) = 0, L(b|bb) = -2$$

— 从等级 $\eta = 0$ 开始，给出向量 (0, 0)，只匹配上密码 **bba** （前驱“aa”匹配等级 0，但是没有转移能匹配等级 0）

— 等级 $\eta = -1$，给出向量 (-1, 0)，匹配 **aba**（前驱“ba”没有后续的转移匹配等级 0）。还有向量 (0, -1)，匹配 **aaa** 和 **aab**。

— 等级 $\eta = -2$，给出三个向量：(-2, 0) 没有匹配（因为没有初始概率能匹配等级 -2），(-1, -1) 匹配 **baa** 和 **bab**，(0, -2) 匹配 **bba**。

2.2 OMEN 算法

之前展示的枚举算法，$enumPwd(\eta, \varphi)$ 用了两个参数。这两个参数需要预先准备，φ（猜测密码的长度）的选择是非常有挑战性的，出现在训练数据当中一种密码长度的出现频率对于一个具体密码长度被猜测的频率不是一个好的参考。比如，我们假设许多密码的长度是 7 和 8，密码长度为 7 的成功概率很大同时搜索空间很小。因此，长度为 7 的密码应该先被猜测。所以，我们使用一个合适的算法，对于每一个长度记录成功的概率然后安排更多的密码来猜测这些长度会更加的高效。

下面我们更加准确的描述我们的合适密码安排算法的流程：

1. 我们设 $\varphi = n$ （我们考虑长度从 3 到 20，比如 $n = 17$），执行 $enumPwd(0, \varphi)$ 并且计算成功概率 $sp_{\varphi,0}$ 。这个概率等于长度为 φ 时，成功的猜测密码在所有生成的猜测密码中所在的比例。
2. 通过成功的概率来建立一个数组长度为 n 的数组 L，其中每一个元素都是一个三元组$(\mathrm{sp}, level, length)$。（第一个元素 $L[0]$ 表示着最大的成功概率）
3. 选择成功概率最高的密码长度，比如 $L[0] = (sp_0, level_0, length_0)$ 然后把它从列表中删除。
4. 执行 $enumPwd(level_0 - 1, length_0)$ ，计算一个新的成功概率 sp^*, 然后添加新的元素 $(sp^*, level_0 - 1, length_0)$ 到数组 L 中。
5. 对数组 L 进行排序然后执行第三步直到 L 变成一个空数组或者有了足够的猜测密码。

2.3 选择参数

在这个部分，我们讨论几种参数的选择以及精确度和性能之前的权衡的必要性。这三个关键的参数是：n 阶算子的规模，字母表的规模和多种等级的枚举密码的规模。

n 阶算子的规模：这个参数对精确性有很大的影响，一个较大的 n 通常会得到更好的结果同时 n 阶算子提供了更加精确的密码分布近似。但是，它需要更长的执行时间，更大的内存和存储需求。训练数据的大小也是一个很重要的，因为只有足够多的数据才能准确的估计参数（比如初始概率和转移概率）。我们评估我们的算法使用 $n = 2,3,4$ ，结果描述在图 1 中。作为预期，更大的 n 得到了更好的结果。我们处理了 5 阶算子的有限实验，结果没有展示在图中，但是比 4 阶算子的结果有一些。总体上，5 阶算子需要更多的运行时间和内存需求，相比少量的精度提升，我们决定使用 $n = 4$。

字母表的规模：字母表的规模也是非常有潜力去极大程度影响攻击的因素，更多的字母表规模以为着更多的参数需要去估计并且运行时间和内存需求的增加。相反，小的字母表意味着不是所有的密码都能被生成。

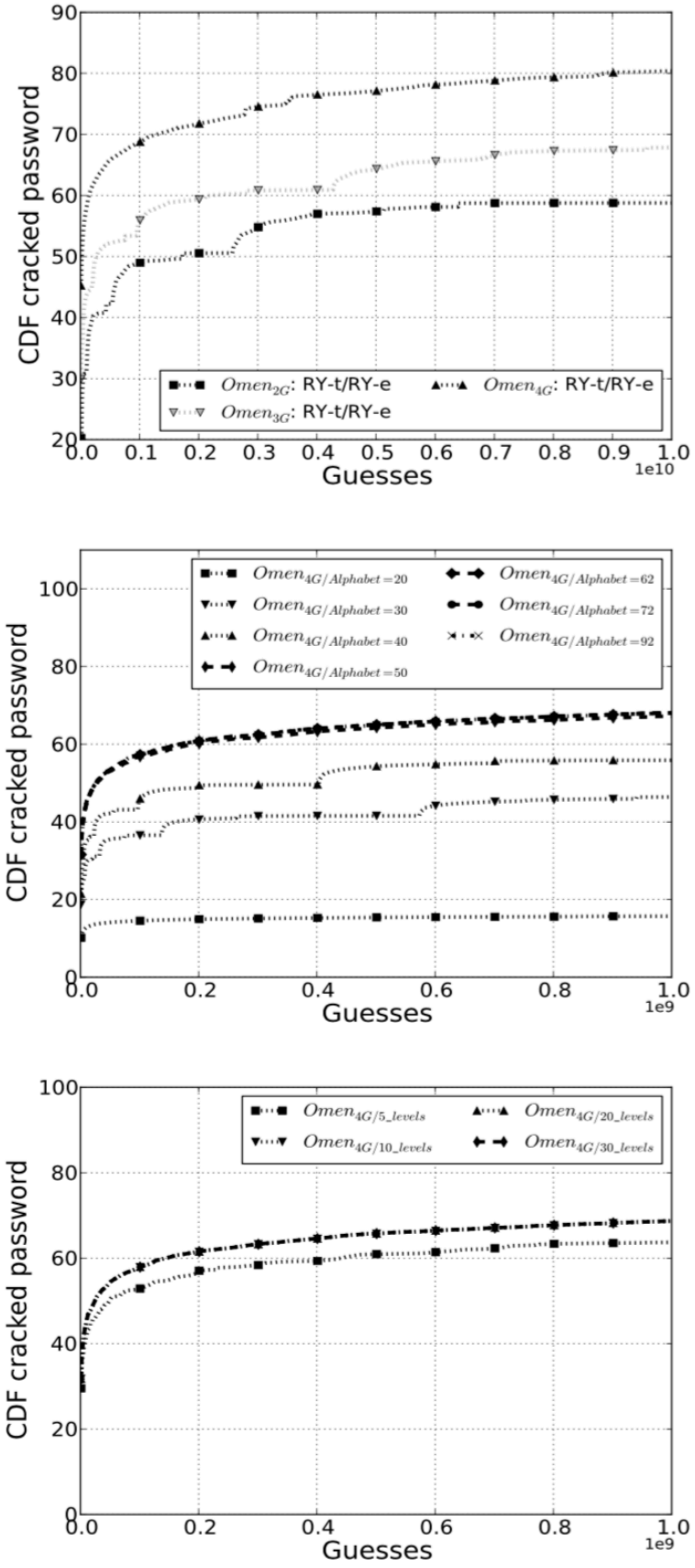

图 1. 对比不同的算子规模（第一幅），字母表规模（第二幅）和不同的等级值（第三幅），使用 RockYou 数据集

我们测试了多种字母表规模通过设置 $k = 20,30,40,50,62,72,92$ ，k 个在训练数据当中出现频率最高的字母组成的字母表，结果在图 1（第二幅）中给出。我们清晰的看到字母表的规模从 20 增长到 62 时，精确度的增长。k 值更高的增长没有为结果带来显著的增长了。主要是因为在 RockYou 数据集当中，用户喜欢使用字母符号来作为密码而不是特殊字符。不考虑数据集，我们设定 72 个字母作为字母表，可以观察到不同语言（字母）的数据集，比如中国拼音，会为 OMEN 设置不同的参数。

等级的数值：第三个重要的参数就是枚举候选密码的等级数值。更高的数值能潜在地增长准确度，但是还是会增加运行时间。实验结果展现在图 1（第三幅）。我们可以看到等级数值从 5 增加到 10 时显著的增长，但是更多的从 20 到 30 的增长并没有为结果带来显著的提升。

被选择的参数：除非在有其他的状况，根据上面的分析，我们选择 4 阶算子，规模为 72 的字母表和数值为 10 的等级来使用 OMEN 算法。

3 评估 OMEN 的性能

在这个部分，我们展示了加强的马尔科夫模型密码攻击和之前的 state-of-the-art 方法的比较。

3.1 数据集

我们评估了在多个数据集下的密码猜测算法。最大的公开的密码集是 RockYou 数据集，包含了 3 千 2 百 6 十万个密码，在 2009 年通过 SQL 注入攻击获得。这个数据集有两个优点：第一，它很大的规模可以让我们训练一个好的马尔科夫模型。第二，它是通过 SQL 注入攻击获得，因此可以减少用户在网站服务上的风险。我们随机的将 RockYou 分割为两个数据集：3 千万数据的训练集（RY-t）和 2.6 百万的测试集（RY-e）。

MySpace 数据集包含了 50 000 密码（因为不同的数据清洗算法或者不同时间点泄漏，不同版本、不同规模的密码都存在）。这个密码在 2006 年通过网络诈骗获得。

Facebook 数据集 2011 年在 pastebin 网站上被公布。这个数据集包含了 Facebook 的密码和相关的邮箱地址。还不知道黑客是如何获取到的数据，但是大概率是通过网络诈骗攻击收集到的。

Algorithm	Training Set	#guesses	Testing Set		
			RY-e	MS	FB
Omen	RY-t	10 billion	80.40%	77.06%	66.75%
	RY-t	1 billion	68.7%	64.50%	59.67%
PCFG [24]	RY-t	1 billion	32.63%	51.25%	36.4%
JtR-Markov [16]	RY-t	10 billion	64%	53.19%	61%
	RY-t	1 billion	54.77%	38.57%	49.47%
JtR-Inc	RY-t	10 billion	54%	25.17%	14.8%

表格 1. 总览 10 亿次（或者特定的 100 亿次）密码攻击实验的测试数据比例

道德的考虑：研究披露密码的数据集可以说是了解用户真实密码实践的最好的帮助，也被一些研究使用 [24,23,5]。同时，这些数据已经是公开给大众的，尽管如此，我们会使用必要的防护手段来对待这些数据，并且我们在发布研究成果时不会涉及到真实的密码(c.f.[7])。

3.2 比较 OMEN 和 JtR 的马尔科夫模型

图二（第一幅）展示了 OMEN 和 JtR 的马尔科夫模型（通过 Narayanan et al 的密码索引方法[16]来实现的）对比。这两个模型都是通过 RockYou 数据集(RY-t)进行训练。然后对于 JtR 的马尔科夫模型，我们调整了猜测次数 T（10 亿或者 100 亿），并且为 T 个输出的密码计算了相关的等级数值(η)。

这个曲线展示了通过我们加强的密码猜测顺序,在攻击速度上有了很大的提高。实际上，JtR 的马尔科夫模型输出的猜测没有特定的顺序，这意味着密码几乎是随机的出现在猜测中。这个行为导致了在图二（第一幅）中几乎线性的曲线。一个可能的问题在 T 以后 JtR 的马尔科夫模型何时会超过 OMEN，答案是不会，因为在 T 点以后 JtR 的马尔科夫模型结果不会再线性的增长了。并且更大的 T 值会导致水平的曲线。为了证明这个结论，我将 T 从 10 亿调整 100 亿做了同样的实验。图三展示了 10 亿次猜想（左）和 100 亿次猜想（右），我们可以看到曲线变成了水平的。

为了展示我们结果的普适性,我们在不同的数据集上比较了破解结果:RY-e,FB 和 MS。排序的优势让 OMEN 在前 9 千次猜测中比 JtR 的马尔科夫模型多破解了 40%的密码（独立于数据集）。并且 JtR 马尔科夫模型需要至少 8 倍的猜测才能达到相同的效果，对于 RockYou 数据集，结果是更加明显的：在前 1 千次猜测中（图三（右））OMEN 破解了 45.2%的 RY-e 密码，JtR 的马尔科夫模型需要至少 70 亿次才能达到相同的结果（$T = 100$ 亿）。

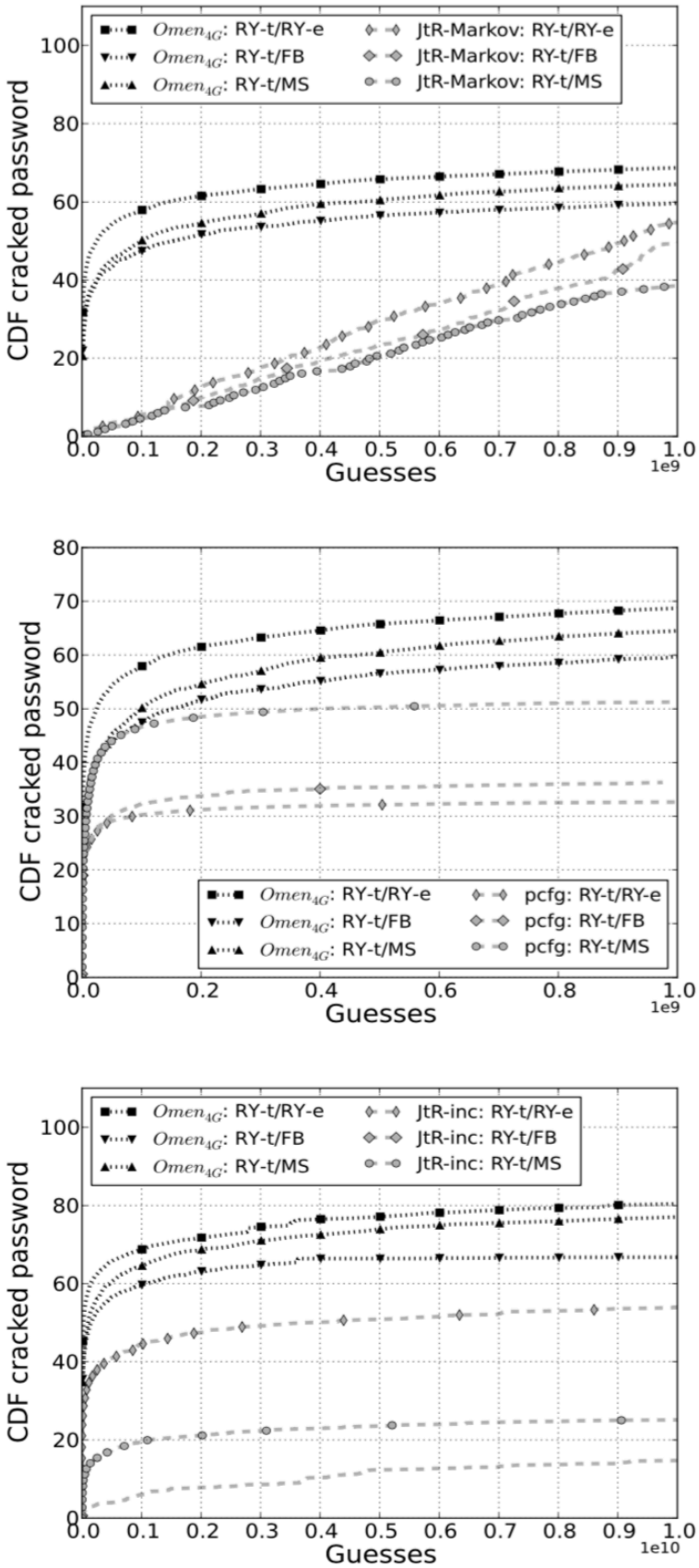

图 2. 比较 OMEN 和 JtR 的马尔科夫模型进行 10 亿次猜测（第一幅），和 PCFG（第二幅），和 JtR 递增模型（第三幅）

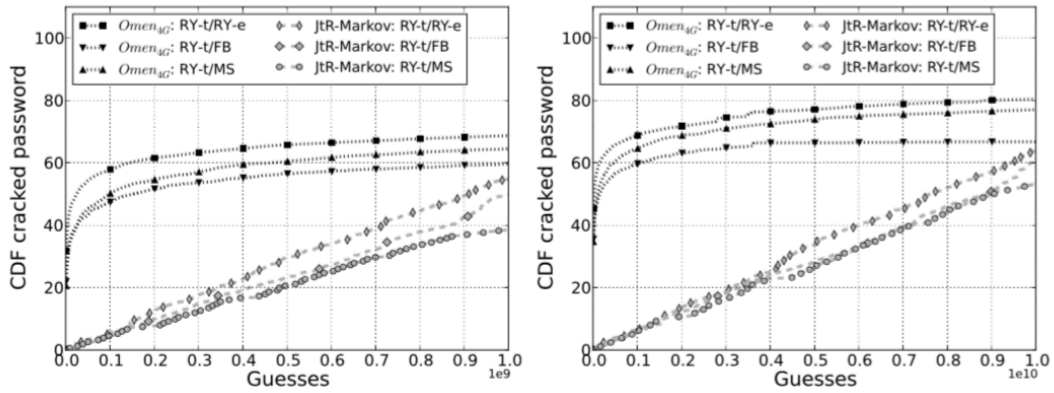

图 3. 比较 OMEN 和 JtR 的马尔科夫模型进行 10 亿次猜测（左）和 100 亿次猜测（右）

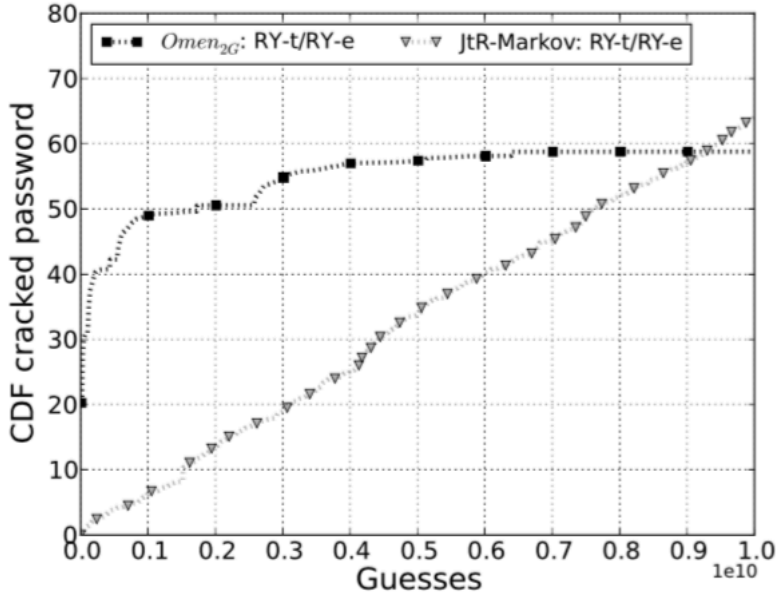

图 4. 比较使用 2 阶算子的 OMEN 和 JtR 的马尔科夫模型

在上诉的对比实验中，OMEN 用了 4 阶算子（c.f.　第 2，3 节），JtR 的马尔科夫模型用了 2 阶算子。能看出有不同的影响，我们提供一种附加的对比试验，OMEN 和 JtR 的马尔科夫模型都使用 2 阶算子。结果和期望的符合，JtR 还是给出了一条直线，这意味着 OMEN 有更好的破解速度。可以从 10 亿猜测 OMEN 破解了 50%，而 JtR 少破解了 10%来看出 OMEN 速度的优势。在 T 点，JtR 停止了增长，两种算法的表现大致相同。能够看到两个模型不是所有参数都相同（比如字母表的规模，等级数值等），我们的破解比例在 T 点只有很小的区别在。

3.3 对比 OMEN 和 PCFG

图 2（第二幅）对比了 OMEN 和 Weir et al.[24]的 PCFG 密码猜测，基于开源的代码[19]。我们使用文字里描述的配置：我们使用 RY-t 来提取出语法和 dict-0294[25]来生成可选的密码。

图 2 展示了 OMEN 比 PCFG 算法效果更好。在 2 亿猜测以后，OMEN 比 PCFG 在 RY-e 和 FB 数据集上多破解了 20%的密码，在 MS 数据集上多破解了 10%。这是非常有趣的能看到数据集多 PCFG 的表现的影响：PCFG 在 MS 上表现的比 FB 和 RY-e 更好。我们相信主要原因是 PCFG 的语法是在 MS 的子集上训练的，所以这个方法更适合去猜测 MS 上的密码。OMEN 大致上在所有数据集中达到了差不多的结果，也证明了扎实的学习阶段。最终，可以看到 PCFG 大致在 3 亿猜测后稳定了下来并且结果很难在提升，而 OMEN 还是能有明显的进步。

3.4 对比 IMEN 和 JtR 递增模型

我们还对比了 OMEN 和 JtR 递增模型，展现在图 2（第二幅）中。和之前的经验相似，两个算法都是在 3 千万的 RockYou 训练集上训练，在 RY-e，MS 和 FB 数据集上测试。可以清晰的看到 JtR 递增模型比 OMEN 的猜测效果要差。

4 论述和总结

在这次工作中，我们展现了高效的基于马尔科夫模型的密码猜测算法（OMEN），它比现在很多公开的密码猜测算法要更加的出色。对于开放的密码数据集，我们发现我们能够在100 亿次的猜测中猜测出 80%的密码。马尔科夫模型以一个高效的密码猜测工具，之前的工作仅仅以算法内部所规定的顺序来输出相关的猜测（和它真实的出现频率相关较少），OMEN 能够大致按照频率递减的顺序输出密码猜测，因此能大幅度的提高真实的密码猜测速度。更进一步，我们展示了很多的实验来估算不同的参数对算法精确度的影响，并且找到了一些理想的参数。我们相信 OMEN 能够被一些组织用来做预防性的测量，来确认他们的用户没有选择安全性较弱的密码。

5 引用

[1] Bishop, M., Klein, D.V.: Improving system security via proactive password check- ing. Computers & Security 14(3), 233–249 (1995)

[2] Bonneau, J.: The science of guessing: analyzing an anonymized corpus of 70 million passwords. In: Proc. IEEE Symposium on Security and Privacy. IEEE (2012)

[3] Bonneau, J., Herley, C., van Oorschot, P.C., Stajano, F.: The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In: Proc. IEEE Symposium on Security and Privacy. IEEE (2012)

[4] Burr, W.E., Dodson, D.F., Polk, W.T.: Electronic authentication guideline: NIST special publication 800-63 (2006)

[5] Castelluccia, C., Dürmuth, M., Perito, D.: Adaptive password-strength meters from Markov models. In: Proc. Network and Distributed Systems Security Symposium (NDSS). The Internet Society (2012)

[6] Dell'Amico, M., Michiardi, P., Roudier, Y.: Password strength: an empirical anal- ysis. In: Proc. 29th conference on Information communications, INFOCOM 2010, pp. 983–991. IEEE Press, Piscataway (2010)

[7] Egelman, S., Bonneau, J., Chiasson, S., Dittrich, D., Schechter, S.: It's not stealing if you need it: A panel on the ethics of performing research using public data of illicit origin. In: Blyth, J., Dietrich, S., Camp, L.J. (eds.) FC 2012. LNCS, vol. 7398, pp. 124–132. Springer, Heidelberg (2012)

[8] HashCat. OCL HashCat-Plus (2012), http://hashcat.net/oclhashcat-plus/

[9] Kedem, G., Ishihara, Y.: Brute force attack on unix passwords with SIMD com- puter. In: Proc. 8th Conference on USENIX Security Symposium, SSYM 1999, vol. 8. USENIX Association (1999)

[10] Kelley, P.G., Komanduri, S., Mazurek, M.L., Shay, R., Vidas, T., Bauer, L., Christin, N., Cranor, L.F., Lopez, J.: Guess again (and again and again): Mea- suring password strength by simulating password-cracking algorithms. In: Proc. IEEE Symposium on Security and Privacy. IEEE (2012)

[11] Klein, D.V.: Foiling the cracker: A survey of, and improvements to, password secu- rity. In: Proc. USENIX UNIX Security Workshop (1990)

[12] Komanduri, S., Shay, R., Kelley, P.G., Mazurek, M.L., Bauer, L., Christin, N., Cra- nor, L.F., Egelman, S.: Of passwords and people: Measuring the effect of password- composition policies. In: CHI 2011: Conference on Human Factors in Computing Systems (2011)

[13] Li, Z., Han, W., Xu, W.: A large-scale empirical analysis of chinese web passwords. In: Proc. 23rd USENIX Security Symposium, USENIX Security (August 2014)
[14] Ma, J., Yang, W., Luo, M., Li, N.: A study of probabilistic password models. In: Proc. IEEE Symposium on Security and Privacy. IEEE Computer Society (2014)
[15] Morris, R., Thompson, K.: Password security: a case history. ACM Communica- tions 22(11), 594–597 (1979)
[16] Narayanan, A., Shmatikov, V.: Fast dictionary attacks on passwords using time- space tradeoff. In: Proc. 12th ACM conference on Computer and communications security (CCS), pp. 364–372. ACM (2005)
[17] OpenWall John the Ripper (2012), http://www.openwall.com/john
[18] The password meter, http://www.passwordmeter.com/
[19] PCFG Password Cracker implementation Matt Weir (2012), https://sites.google.com/site/reusablesec/Home/ password-cracking-tools/probablistic_cracker
[20] Provos, N., Mazières, D.: A future-adaptive password scheme. In: Proc. Annual Conference on USENIX Annual Technical Conference, ATEC 1999. USENIX As- sociation (1999)
[21] Schechter, S., Herley, C., Mitzenmacher, M.: Popularity is everything: a new ap- proach to protecting passwords from statistical-guessing attacks. In: Proc. 5th USENIX Conference on Hot Topics in Security, pp. 1–8. USENIX Association (2010)
[22] Spafford, E.H.: Observing reusable password choices. In: Proc. 3rd Security Sym- posium, pp. 299–312. USENIX (1992)
[23] Weir, M., Aggarwal, S., Collins, M., Stern, H.: Testing metrics for password cre- ation policies by attacking large sets of revealed passwords. In: Proc. 17th ACM Conference on Computer and Communications Security (CCS 2010), pp. 162–175. ACM (2010)
[24] Weir, M., Aggarwal, S., de Medeiros, B., Glodek, B.: Password cracking using probabilistic context-free grammars. In: Proc. IEEE Symposium on Security and Privacy, pp. 391–405. IEEE Computer Society (2009)
[25] Word list Collection (2012), http://www.outpost9.com/files/WordLists.html