高并发弱密码分层检测系统设计与实现

吴 颖

南京传媒学院 江苏 南京 211172

【摘 要】：对于入侵防御系统千万级弱密码库检测时的内存占用大、更新过程中阻塞业务的问题，提出了一种分层异构存储弱密码的检测系统架构。系统分为两层过滤：前端布隆过滤器初步过滤出非弱密码查询，后端分热-温-冷三层存储（压缩双数组 Trie、分片的布谷鸟过滤器、NVMe 驻留位图）实现准确匹配。为方便动态更新，引入基于写时复制的分片替换方案实现在更新过程中不中断服务。基于 WeakPass 数据集包含 1000 万条弱密码的测试发现，相比 dense_hashMap 基线的弱密码检测，系统内存使用量从 1960 MB 降到了 628 MB；同时系统在线测试支持 100 kQPS 的负载，平均查询延迟 3.9 μs、单核吞吐 138.5 kQPS；5 万条弱密码的动态数据增量更新耗时 11.8 ms 且零停机。

【关键词】：弱密码检测；分层存储；布隆过滤器；布谷鸟过滤器；增量更新；高并发系统

DOI:10.12417/2811-0722.26.09.049

1 引言

弱密码检测是入侵检测（Intrusion Detection System，IDS）、身份认证系统、云服务商安全管理系统等自动化安全系统的常用功能模块之一。将用户输入的密码与预设弱密码库进行快速比对，能够及时有效地阻止暴力破解、撞库攻击、凭据填充等攻击行为。林国飞等[1]针对 MySQL 数据库进行了弱密码检测的研究，解析数据存储文件获取密码信息，并根据 MySQL5.6、5.7、8.0 不同版本，对密码实行明文匹配或者加密摘要匹配，提高了数据库环境下的弱密码检测效果和效率。沈舒莉等[2]研究了容器环境下的弱密码检测方法、检测系统，提升了云原生环境下弱密码检测的效率。

随着互联网上人数的增加，弱密码库已达到千万甚至亿级的规模[3]，10Gbps 以上高并发数据对检测系统的实时性、检测资源的利用率、动态更新能力的要求提也越来越高。目前的弱密码检测方法大多使用了哈希表、标准 Trie 树或布隆过滤器中的一个来实现，实际使用中普遍存在如下 4 个方面的问题：（1）内存过大，哈希表混入 Trie 树在千万级弱密码规模下的内存占用可达数 GB，不适用于边缘设备和物联网终端；（2）检测性能，全量遍历匹配哈希表的方法单核吞吐通常低于 10 kQPS；（3）更新时阻塞，更新时全局锁机制导致业务中断和带来的时延抖动；（4）资源浪费，未注意到其“非弱密码比例偏大、弱密码访问呈偏态”的特点做专项优化。

针对上述缺陷，国内外学者从不同方面进行了优化研究。文献[4]研究面向安全运维的弱密码告警偏差智能补偿检测方法，提高检测准确率。何乐生等[5]研究面向物联网的弱密码检测系统轻量化框架。刘艳娇等[6]研究压缩 Trie 树的建模与理论分析，为快速串匹配提供计算职称。王瀚橙等[7]研究无锁并发表的布谷鸟过滤器，提高高并发请求下的成员查询性能。现有研究大都各部分做了优化，缺少将前置过滤、分层匹配、增量更新、资源调度融综合在一起的优化思路。

2 相关工作

弱密码检测技术经过多年发展，已形成基于规则匹配、机器学习、深度学习及网络流量分析等多种技术路线。本节从特征提取方式、应用场景和系统架构三个维度对现有研究进行分析，并定位本文工作创新点。

2.1 基于规则与形态学特征的检测方法

传统弱密码检测一般采用规则匹配和启发式策略。栗会峰等[8]针对电力系统，在形态学上提出了弱密码的深度学习检测方案。该方案将口令转化成 28×28 的图像，将口令的形态特征用卷积神经网络进行学习，有效识别出存在键盘坐标分布规律的弱密码。实验结果表明该方案的准确率相比于 N-gram 马尔科夫链模型和卡巴斯基评估器更高。但该方法仅适用于行业上的（如电力系统）定制化需求，并且深度学习模型计算成本高，不能满足 10 Gbps 级的实时检测要求。Maralbayev 等[9]提出了一种基于用户个人信息（名字、姓氏、出生年月等的组合）进行弱密码检测的算法。该方法可以构建易受到暴力破解的密码字典，随着新用户信息的增加字典不断被扩充，在前端应用软件实时报警弱密码。Rzayeva 等[10]扩充了该思路，构造利用社交媒体和个人信息的密码易被猜测的密码检测方法。这类方法具有发现有个人因素语义弱密码的实用价值，但也有不足之处：一种事依赖于用户上报额外的信息，在入侵检测系统等被动检测场景下难以采集；另外一种是实时动态字典的扩充会线性增长空间和计算代价，缺乏针对大规模密码库字典的的分层优化机制。

作者简介：吴颖（1984.9-），就职于南京传媒学院，计算机科学与技术系，高级工程师。
研究方向：高性能网络流量处理，网络安全，软件工程。

2.2 面向物联网场景的检测框架

随着 IoT 设备的普及，针对 IoT 设备的弱密码检测也越来越成为研究热点。Qu[11]设计了一套针对 IoT 设备 Web 应用程序的弱密码自动检测的框架，并在北京、山东、浙江三省的广域网上开展了验证，发现存在弱 Web 密码的 IoT 设备共 12,179 台（占比 7.58%），揭示了大规模 IoT 设备里弱密码的普遍存在性，但 Qu 的方法仍然是主动扫描搜索式检测方式，检测时延较大（分钟级），没有解决千万量级密码库匹配时的内存炸裂问题。

2.3 现有研究的共性与本文工作

综合以上几种研究工作，目前可用的弱密码检测方法主要存在以下缺点：（1）存储单一化。各方法中密码库的存储结构几乎都以哈希表、标准 Trie 树或者单一布隆过滤器存储密码，在千万量级时内存消耗可达几个 GB[8,9,10]，无法满足边缘计算节点和物联网设备的算力要求。（2）检测吞吐不足。基于深度学习的方法[9]推理延迟在毫秒量级，基于动态字典扩展的方法[10]随着字典长度增长，查询延迟也会线性增加，这些方法均无法保证高并发情况下 μs 检测延迟要求。（3）更新操作阻塞。目前方案采用全局锁的方式处理密码库更新[10,11]，更新时会造成服务阻塞与服务延时，不能满足 7×24 小时全时守护要求。

对于以上问题，提出一种基于前置过滤+分层压缩式的弱密码自动化检测方法。与已有的工作相比，主要贡献包括：（1）用热-温-冷三层异构存储结构替代单一存储结构，降低内存消耗、保证查询性能；（2）采用全局优化布隆过滤器与分层匹配机制，将单核吞吐提升至 138.5 kQPS，较传统方案提升一个数量级；（3）设计影子分片与原子指针切换机制，实现增量更新零停机，解决了更新阻塞问题。

3 系统设计与方法

针对第 2 节所述现有方法的局限性，设计了一种分层异构存储的检测架构。系统整体架构如图 1 所示，主要由初始化模块、前置过滤模块、分层匹配模块、增量更新模块与动态资源调配模块构成。

系统初始化阶段，后台线程并行构建热-温-冷三层压缩存储结构与全局优化布隆过滤器。用户密码输入后，先经布隆过滤器完成前置过滤，快速排除非弱密码；对疑似弱密码，按热、温、冷层级依次匹配，匹配成功输出告警，否则判定为安全。增量更新模块通过影子分片与原子指针切换实现无感知更新；动态资源调配模块根据实时访问频率完成层级调整与冷数据迁移。系统同时支持快照持久化与日志重放，保障故障恢复能力。

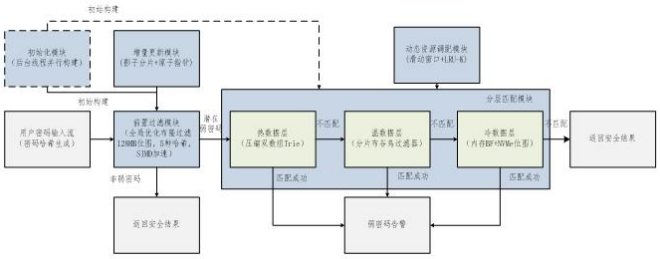

图 1 系统整体架构图

3.1 全局优化布隆过滤器前置过滤

前置过滤是提升整体检测效率的关键环节，目标是在可控内存开销下剔除 90%以上的非弱密码，减轻后续匹配压力。全局布隆过滤器采用 128MB 位数组，采用 MurmurHash3、FNV1a、CRC32、Jenkins、CityHash 共 5 种哈希函数。位数组按 64 字节缓存行对齐，降低缓存未命中损失；通过 AVX-512 指令集实现 SIMD 并行哈希值计算，提升初始化与过滤吞吐。

前置过滤执行流程：

1.对输入密码并行计算 5 组哈希值，存入 CPU L1 缓存；

2.将哈希值映射到位数组对应位置；

3.若任意位置值为 0，直接判定为非弱密码；

4.若全部位置为 1，判定为潜在弱密码，将哈希值传递至分层匹配模块复用。

该模块吞吐量可达 500kQPS 以上，误报率控制在 0.01%以内，显著降低后续计算压力。

3.2 热-温-冷三层分层匹配

分层匹配遵循“逐级检查、高频优先、流水线并行”的原则，复用前置过滤得到的哈希值，依次在热、温、冷层执行匹配，命中则立即返回结果，减少平均响应时间。

（1）热数据层匹配：热数据层存放访问频率前 20%的高频弱密码，采用压缩双数组 Trie（CDA-Trie）结构。Base 数组初始长度 1024，按 2 的幂次动态扩容；Check 数组长度 4096，单元素 12bit（8bit 字符值+4bit 状态转移地址）。高频节点采用 4bit 压缩状态机存储状态偏移量，低频节点保留 12bit 完整状态，在保证速度的同时降低内存。匹配时采用双栈回溯算法：时间复杂度 O(m)，m 为密码长度。

（2）温数据层匹配：温数据层存放访问频率 20%~50%的中频弱密码，采用分片布谷鸟过滤器阵列。通过 MurmurHash3 生成 64 位哈希值，按哈希区间划分为 N 个分片，单个分片包含 10 万条目、4 个哈希桶、每桶存储 2 个 8bit 指纹。查询采用 CAS 指令实现无锁并行，以轮询方式均衡多核负载，结果存入 L1 缓存提升重复查询效率。匹配成功则判定为弱密码，否则就进入冷数据层。

（3）冷数据层匹配：冷数据层存放访问频率最低的 50%

弱密码，采用“内存布隆过滤器预判+NVMe 磁盘位图”混合架构。内存布隆过滤器为 1MB 位数组，采用 FNV1a、Jenkins、CRC32 三组哈希完成存在性预判；若预判不存在，直接判定安全；若预判可能存在，则访问 NVMe 磁盘 Roaring Bitmap，并以 AVX-512 指令集加速计算。查询结果使用 LRU 缓存，减少重复磁盘 I/O。

3.3 影子分片与原子指针增量更新

为解决传统更新阻塞问题，采用影子分片与原子指针切换机制，所有更新操作在后台影子分片完成，不影响前台检测业务。增量更新流程如图 2 所示。

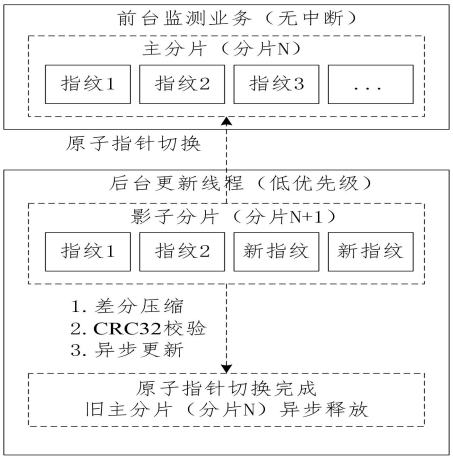

图 2 增量更新流程示意图

增量更新步骤：

1.后台低优先级线程将新增弱密码写入影子分片，以差分压缩记录差异，通过 CAS 保证原子性；

2.使用 CRC32 校验影子分片数据完整性；

3.校验通过后，异步将新增密码哈希插入全局布隆过滤器；

4.通过 std::atomic 内存屏障实现微秒级原子指针切换，将主分片指向影子分片；

5.将旧主分片标记为可回收，后台线程在确认无活跃引用后异步释放内存。

该机制实现无感知动态更新，更新期间不中断业务、不抬升检测时延。

3.4 动态层级调整与冷数据迁移

以 60s 为滑动时间窗口实时统计访问频率，按频率划分热（前 20%）、温（20%~50%）、冷（低于 50%）三层；采用 LRU-K（K=2）识别超过 24 h 未访问的冷数据；后台线程迁移至 NVMe 磁盘 Roaring Bitmap。迁移前通过 CAS 锁定分片保证线程安全，迁移完成后将内存归还内存池，降低内存占用与碎片。该策略使存储资源与实际访问模式动态匹配，避免静态分层造成浪费。

4 实验与分析

4.1 实验设置

（1）实验数据集：实验采用 WeakPass 公开弱密码数据集（weakpass_4.txt）[12]，该数据集整合多源泄露密码，包含约 20 亿条记录，时效性与覆盖度优于传统数据集。为保证可复现性，选取去重后前 1000 万条高频密码作为测试集，并按访问频率划分为热（200 万）、温（300 万）、冷（500 万）三层，再选取 5 万条去重密码作为增量更新测试集。

（2）实验环境：实验硬件：Intel Xeon Gold 6230R（26 核，2.1GHz），96GB DDR4 内存，1.2TB NVMe SSD；软件环境：Ubuntu 20.04LTS,GCC 11.2,C++17,OpenMP 5.0。

（3）对比方案与评价指标：对比方案选取三类主流方法：

1.哈希表（HT）：采用 C++STL 的 dense_hash_map 存储弱密码哈希值，通过哈希表查找实现匹配；

2.标准 Trie 树（STT）：采用标准双数组 Trie 树存储弱密码，通过顺序遍历实现匹配；

3.布隆过滤器+标准 Trie 树（BF+STT）：传统前置过滤+全量 Trie 匹配。

评价指标：

1.内存占用（MO）：总内存开销，单位 MB；

2.平均查询延迟（AQL）：单条检测耗时，单位 μs；

3.单核吞吐量（SCT）：单 CPU 核心每秒处理条数，单位 kQPS；

4.更新延迟（UL）：5 万条增量更新平均耗时，单位为 ms。

4.2 实验结果

（1）内存占用对比：表 1 为 1000 万条密码下各方案内存开销。本文方法仅 628 MB，较哈希表降低 67.9%，较标准 Trie 树降低 52.9%，较 BF+STT 降低 27.3%。主要原因是三层压缩结构与冷数据磁盘卸载显著降低内存占用。

表 1 各方案内存占用对比

方案	内存占用（MB）
哈希表(HT)	1960
标准 Trie 树(STT)	1332
布隆过滤器+标准 Trie 树(BF+STT)	864
本文方法	628

（2）平均查询延迟对比：如表 2 所示，100 kQPS 压力下，本文方法平均查询延迟仅 3.9 μs，较 HT 降低 86.4%，较 STT 降低 84.2%，较 BF+STT 降低 75.8%。得益于前置过滤大量剔除无效查询、流水线并行与哈希值复用。

表 2 平均查询延迟对比（100k QPS）

方案	平均查询延迟(μs)
哈希表(HT)	28.6
标准 Trie 树(STT)	24.7
布隆过滤器+标准 Trie 树(BF+STT)	16.1
本文方法	3.9

由表 2 可知，本文方法平均查询延迟为 3.9μs，理论单核峰值吞吐约为 256 kQPS。实测吞吐 138.5kQPS 约为理论值的 54%，主要源于多线程并行时的缓存一致性同步开销、内存访问延迟抖动及 OpenMP 调度损耗。

（3）单核吞吐量对比：如表 3 所示，本文方法单核吞吐达 138.5 kQPS，为哈希表的 15.7 倍、标准 Trie 树的 13.2 倍、BF+STT 的 2.79 倍。AVX-512 加速与无锁并行设计使全流程得到充分优化。

表 3 单核吞吐量对比

方案	单核吞吐量（kQPS）
哈希表(HT)	8.8
标准 Trie 树(STT)	10.5
布隆过滤器+标准 Trie 树(BF+STT)	49.7
本文方法	138.5

（4）增量更新延迟对比：如表 4 所示，5 万条增量更新场景下，本文方法更新延迟仅 11.8 ms，较 HT 降低 91.9%，较 STT 降低 89.1%，较 BF+STT 降低 77.1%，且全程不阻塞前台检测。

表 4 增量更新延迟对比（5 万条）

方案	更新延迟（ms）
哈希表(HT)	145.6
标准 Trie 树(STT)	108.3
布隆过滤器+标准 Trie 树(BF+STT)	51.6
本文方法	11.8

4.3 讨论

实验结果表明，所提方法在内存占用、查询时延、检测吞吐、更新时延四项核心指标上均显著优于传统方案，能够在大规模、高并发场景下实现低开销、低延迟、无阻塞的弱密码实时检测。

4 结束语

针对高并发网络环境下传统弱密码检测方法存在的内存膨胀、吞吐不足、更新阻塞等问题，提出基于前置过滤与分层压缩的自动化检测方法。通过全局优化布隆过滤器实现高效前置过滤，热-温-冷三层压缩存储实现分级匹配，借助影子分片与原子指针实现无锁增量更新，结合滑动窗口与 LRU-K 完成存储动态调度。实验验证表明，该方法在多项性能指标上均有显著提升，可满足入侵防御、身份认证等高并发场景的安全检测需求。未来将进一步基于 GPU/CUDA 优化并行哈希计算，探索异构平台硬件加速方案，提升极端流量下的检测稳定性。

参考文献：

[1] 林国飞.MySQL 数据库弱密码检测方法和系统,电子设备及介质[P].CN118070269A,2024-05-24.

[2] 沈舒莉,林顺东,施纯毅,等.一种容器内应用服务的弱密码检测方法和系统[P].CN115795440A,2022-12-07.

[3] 腾讯云公共互联网反网络钓鱼专栏.超大规模密码泄露事件下的身份认证安全重构[EB/OL].https://cloud.tencent.com/developer/article/2593528.

[4] 中南大学.一种面向安全运维的轻量级弱密码告警偏差智能修正方法[P].CN121486067A,2026-02-06.

[5] 何乐生,冯毅,岳远康,等.针对物联网设备的旁路攻击及防御方法的研究[J].通信学报,2025,46(2):166-175.

[6] 刘艳娇,万亮亮,王昌晶.压缩特里树的建模与分析[J].江西师范大学学报(自然科学版),2025,49(2):123-129.

[7] 王瀚橙,陈志鹏,戴海鹏,等.无锁并发布谷鸟过滤器[J].软件学报,2025,36(7):3339-3357.

[8] 栗会峰,李铁成,姚启桂,等.基于形态学的电力系统弱口令深度学习检测方案[J].计算机应用与软件,2024,41(10):379-385.

[9] Maralbayev A,Omar G S,Abdulayeva M,et al.New Algorithm of Weak Password Detection[C]//2023 IEEE International Conference on Smart Information Systems and Technologies.Nur-Sultan:IEEE,2023:1-6.

[10] Rzayeva L,Wang Y,Ryzhova A,et al.Password Vulnerability Detection Based on Social Media and Personal Information[C]//2025 IEEE/ACIS 29th International Conference on Software Engineering,Artificial Intelligence,Networking and Parallel/Distributed Computing(SNPD).Bangkok:IEEE,2025:81-86.

[11] Qu J.Research on Password Detection Technology of IoT Equipment Based on Wide Area Network[J].ICT Express,2022,8(2):213-219.

[12] WEAKPASS.weakpass_4.txt[DB/OL].https://weakpass.com/weakpass,2024-01-15.