Computer Science and Application 计算机科学与应用, 2023, 13(7), 1409-1419
Published Online July 2023 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2023.137139

Hans汉斯

基于传统公钥基础设施的密钥协商性能优化研究

吴　凡[1,2]，侯凌燕[1]，张　伟[1,3]

[1]北京信息科技大学计算机学院，北京
[2]北京信息科技大学北京未来区块链与隐私计算高精尖创新中心，北京
[3]北京信息科技大学国家经济安全预警工程北京实验室，北京

收稿日期：2023年6月17日；录用日期：2023年7月14日；发布日期：2023年7月24日

摘　要

传统基于公钥基础设施的密钥协商算法在支持完美前向保密时，需要频繁的密钥协商，且协商步骤复杂，还使用了耗时的非对称密钥算法，性能消耗严重，不适用于资源有限的嵌入式设备中。针对以上问题，本文优化了传统公钥基础设施密钥协商方案，密钥信息不需要频繁通过网络交换，而是双方在本地动态生成密钥，保证了每次加密密钥不同，支持了完美前向保密的特性。实验结果表明，优化后的密钥协商方案性能相比于基于传统公钥基础设施的密钥协商方案显著提升。

关键词

公钥基础设施，密钥协商，完美前向保密

Research on Performance Optimization of Key Agreement Based on Traditional Public Key Infrastructure

Fan Wu[1,2], Lingyan Hou[1], Wei Zhang[1,3]

[1]Computer School, Beijing Information Science & Technology University, Beijing
[2]Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, Beijing Information Science & Technology University, Beijing
[3]Beijing Laboratory of National Economic Security Early-Warning Engineering, Beijing Information Science & Technology University, Beijing

Received: Jun. 17th, 2023; accepted: Jul. 14th, 2023; published: Jul. 24th, 2023

文章引用: 吴凡, 侯凌燕, 张伟. 基于传统公钥基础设施的密钥协商性能优化研究[J]. 计算机科学与应用, 2023, 13(7): 1409-1419. DOI: 10.12677/csa.2023.137139

Abstract

When the traditional key agreement algorithm based on public key infrastructure supports perfect forward secrecy, frequent key agreement is required, and the negotiation steps are complicated. It also uses a time-consuming asymmetric key algorithm, which consumes a lot of performance and is not suitable for embedded devices with limited resources. In response to the above problems, this paper optimizes the traditional public key infrastructure key agreement scheme. The key information does not need to be frequently exchanged through the network, but the two parties dynamically generate the key locally, which ensures that the encryption key is different each time, and supports perfect forwarding characteristic of secrecy. The experimental results show that the performance of the optimized key agreement scheme is significantly improved compared with the key agreement scheme based on traditional public key infrastructure.

Keywords

Public Key Infrastructure, Key Agreement, Perfect Forward Secrecy

1. 引言

在密码学中密钥(key)本质是一串数字序列，是加解密运算参数，是管理秘密信息的钥匙，在明文转化为密文或是密文转化为明文时需要输入对应的密钥才能完成运算。密钥在密码学领域中占着很重要的地位，密钥在密码系统中需要严格的保管，一旦密钥泄露可能会造成严重的数据泄露事件。1883 年 Kerckhoffs 提出了柯克霍夫原则(Kerckhoffs' Principle) [1]，是当今密码学领域中基本原则之一，意思是各种密码算法应该是允许向外公开的，不能影响数据安全性，只要是用其加密的密钥没有发生泄露，密文信息不应该被破解。

在数据传输过程中双方需要保证加密密钥一致，才能计算出正确的加解密结果，因此需要进行密钥协商，密钥协商指的是需要通信的双方在公开的网络信道中通过数据交换得到相同加密密钥的过程。在密钥协商中支持完美前向保密(Perfect Forward Secrecy, PFS) [2]特性，对密钥的安全性保护具有重要的意义，这意味着当前密钥如果发生泄露并不会影响之前的密钥安全性，保证了之前数据的安全。PFS 概念在 IEEE 1363-2000 标准里加入到多种密钥共享协议中。在 2014 年谷歌安全部门和其它安全公司发现了 openssl 标准库中存在安全漏洞，命名为“心血漏洞”，不支持前向保密的网络站点存在严重的数据泄露风险，此漏洞的提出影响了中国几万台服务器，PFS 受到了越来越多人的重视，多家公司对 PFS 提供了支持，Google 在 Gmail、Google Docs 和加密搜索服务中提供了 PFS，Twitter 也为用户支持了 PFS，当前在所有使用 HTTPS 的站点中，80%以上都一定程度上支持了 PFS，可见对 PFS 的支持有着重要的意义。

针对传统基于公钥体系设施(Public Key Infrastructure, PKI)密钥协商方案在支持 PFS 时的性能较低，不适用于资源受限的嵌入式设备的问题，对传统 PKI 密钥协商方案进行了优化，密钥信息不需要频繁通过网络交换，而是同时在本地动态生成密钥，避免使用了耗时的非对称算法，保证了每次加密密钥不同，支持了 PFS 特性，最后通过实验测试，表明本方案相比于传统 PKI 密钥协商方案显著提升。

2. 相关工作

2.1. 密钥协商技术研究现状

Li 等人[3]提出了第一个称为 iTLS 的轻量级安全传输协议，支持完美前向保密特性，并提供无证书的隐式相互认证，iTLS 在收到服务器响应之前动态生成基于身份的早期密钥，允许客户端发送加密数据而无需额外的往返行程。杨鹏飞等人[4]结合了 ECC 椭圆曲线(Ellipse Curve Cryptography, ECC)、三因子认证和中国剩余定理提出了在工业互联网的场景下一对多的密钥协商方案，安全性和性能均一定程度上的提升。龚成等人[5]提出了一种面向车载自组网的基于密钥协商的隐私保护认证方案，此方案利用树的数据结构实现了密钥协商中的密钥生成与更新。夏涛等人[6]设计了一种基于 SM2 的应用于军用无人机的轻量级密钥协商机制，能够有效地应对军用无人机网络所面临的威胁，并且在计算开销方面取得了优势。黄晓晖等人[7]提出了一种基于群签名密钥协商的多方完整性验证方案，此方案应用了区块链技术，将需要保证完整性的数据上传到链上，并且不断将更新的数据上链。王华华等人[8]提出了一种将奇偶校验码与ECC椭圆曲线算法相结合的密钥协商方案，解决无线网络中物理层密钥技术生成的密钥不一致的问题，采用了 ECC 加密算法传递简短密钥块进行密钥协商。张萌楠[9]对现有基于无证书的密钥协商协议进行了分析，存在安全缺陷，针对安全缺陷提出了轻量的无证书密钥协商方案，该方案更具有安全性，更高效，适用于物联网的场景。李等人[10]提出了一种新的密钥协商和身份认证方案 IoT-AKA，是一种基于椭圆曲线的轻量认证机制和方案，该方案基于 eCK 的安全模型，将安全性依赖于椭圆曲线离散对数难题假设和 CDH 困难假设上。

2.2. 完美前向保密技术研究现状

Yang Zheng 等人[11]为 IIoT 系统提出了一种支持 PFS 的高效 AKA 协议，该协议是基于新的动态身份验证凭证(DAC)框架开发的，不使用任何公钥加密原语，性能比当前支持 PFS 的最先进的基于 DAC 的 AKA 协议更快。Syed 等人[12]提出了具有随机初始化向量和较强特性的密钥交换协议，该算法的安全性与 OTP (One Time Pad)算法相同，具备支持 PFS 的能力，提供了一种并行实现算法，可以用更少的时钟周期实现高通量，算法将初始化向量随机化，避免了误用攻击、中间人攻击、重放攻击等。Avoine 等人[13]在对称密钥设置中提出了一种经过身份验证的密钥交换协议，可确保完美的前向保密性，证明该协议是可靠的，并提供其安全性的正式证明。Saman 等人[14]提出了一个密钥管理 SIKM 框架，SIKM 框架包含了密钥协商方案，在每个区域中使用服务器下发解释器，各个客户端在互相进行通信时，通过解释器生成密钥，密钥一段时间更新一次，这种密钥协商方案在性能上与 PKI 密钥协商相比有了明显的优化，但是安全上存在缺陷，源文件并没有得到有效的保护，并且没有完全支持前向保密性。Qing [15]等人认为 Avine 等人密钥保密性设计具有缺陷，并针对物联网环境，提出了 SAKE*协议，其中包含两种类型的密钥主密钥和演化密钥保证了 PFS，SAKE*协议中仅使用了伪随机函数和消息认证码操作来实现身份认证、密钥交换和消息的完整性，提高了密钥协商算法效率，增强了安全性，但是每次协商过程中需要向权威结构去授权，这部分存在通信开销，因此在实际情况下不一定保证性能可以得到优化。

3. 基于 PKI 密钥协商

3.1. 基于 PKI 密钥协商流程

基于 PKI 体系数据加解密与身份认证流程如图 1 所示，用户 A 向 B 安全的发送数据信息。

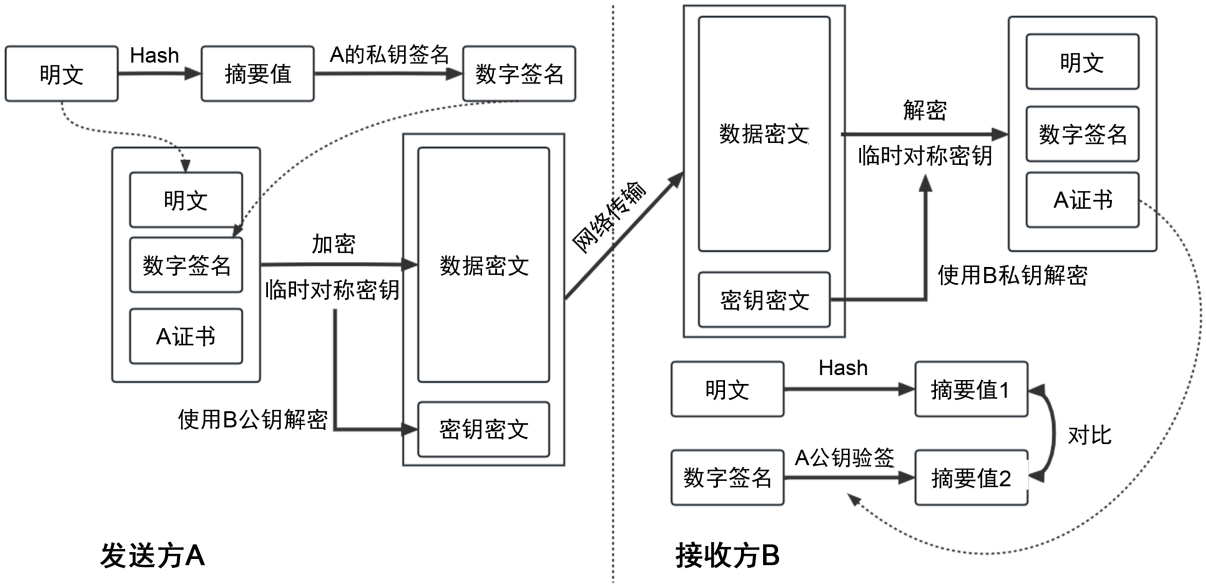

Figure 1. Process diagram of PKI key negotiation
图 1. PKI 密钥协商过程

用户 A 在发送数据时过程如下：

1) 明文通过 Hash 函数计算得到对应的 hash 值；

2) 使用 A 的私钥对明文 hash 值进行签名运算生成 A 的数字签名；

3) 将明文、数字签名和用户 A 的证书打包；

4) 用户 A 随机生成对称密钥作为本次会话密钥，使用会话密钥将 3)中数据包进行对称加密，生成数据密文；

5) 使用用户 B 的公钥将本次会话密钥进行非对称加密得到会话密钥密文；

6) 用户 A 将数据密文和会话密钥密文通过网络发送给用户 B。

用户 B 在接收用户 A 数据时过程如下：

1) 用户 B 使用私钥对会话密钥密文进行解密，得到会话密钥明文；

2) 使用会话密钥明文对数据包密文进行对称解密，得到数据包明文，用户 B 在此步骤可得到用户 A 发送的明文数据；

3) 用户 B 拿到用户 A 的证书，使用用户 A 证书中的公钥字段对 A 的数字签名进行验证，用于鉴别发送者 A 的身份，同时解出用户 A 在步骤 1)中生成的明文 Hash 值；

4) 用户 B 对步骤 2)算得的明文进行 Hash 运算；

5) 对比步骤 3)中验签生成 hash 值和步骤 4)中通过明文 Hash 运算生成的 hash 值，用于鉴别数据的完整性。

使用这种方式发送数据能够有效地对用户进行身份认证，有效地保护数据的完整性，能够有效地防止恶意用户对身份的伪造，极大地保护了网络数据安全。

3.2. PKI 密钥协商存在的问题

在双方设备进行加密通信时，通常使用基于 PKI 的非对称加解密算法进行密钥协商，首先认证对方发送的证书，判断其是否在可信任中心注册过，若没有注册过则认为发送方身份不可信，存在危险隐患。证书认证成功后，使用对方公钥进行验签操作，使用私钥对密钥密文解密得到会话密钥，此过程需要很

大的空间开销和计算开销。首先在密钥协商的过程中，携带的会话密钥密文通常仅为 16 B，但是为了身份认证，发送方需要额外发送自己的数字证书，这些数字证书大小相比于会话密钥信息多出 128 倍，平均 2 K 大小，接收方收到证书信息后需要存储到内存上，并提取其中的公钥数据，因此该密钥协商算法需要很大的空间开销。其次，此过程占用了很大的计算开销，接收方收到发送方数据包时需要验证发送方的数字证书，验证过程需要向认证中心请求服务，存在网络通信开销，认证完成后，还需要多次的非对称密码运算，由于非对称算法计算性能普遍相对较低，基于 ECC 椭圆曲线非对称算法存在较为耗时点乘运算，SM9 存在双线性对运算，比点乘运算更为耗时，所以又存在了较高的时间开销。基于 PKI 的密钥协商算法时间和空间开销相对较高。如果该密钥协商算法支持 PFS 特性，具体流程如图 2 所示，A 向 B 发送数据时每一次都需要身份认证、密钥协商，这部分性能开销较大，随后再加密数据和发送密文，这种性能开销对于资源有限的嵌入式设备来说影响较为明显。

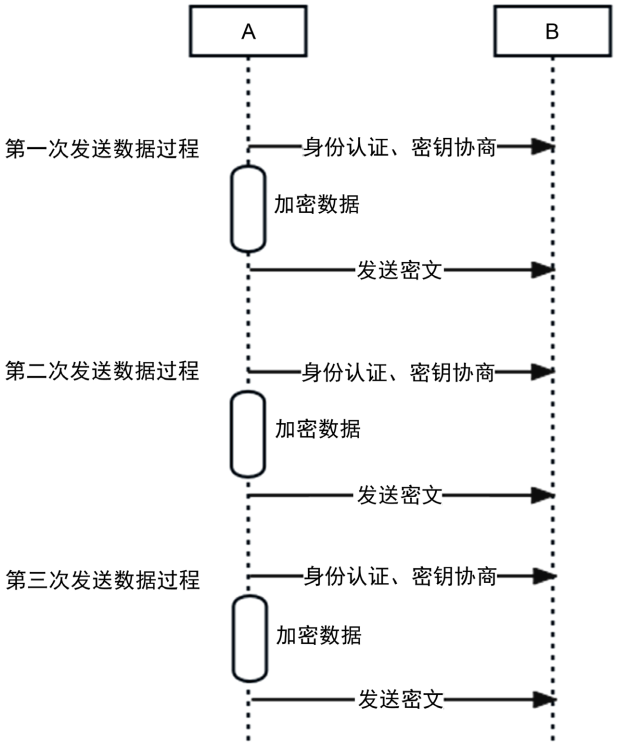

Figure 2. Process diagram of data transmission from A to B in PKI key negotiation scheme
图 2. PKI 密钥协商方案时 A 向 B 传输数据过程

4. 基于 PKI 的密钥协商优化设计

4.1. 优化后密钥协商流程

优化后密钥协商流程总共分为两个阶段，第一阶段是建立连接阶段，用于同步密钥参数，第二阶段是密钥生成阶段，第二阶段随着双方的每一次通信而进行，双方动态的生成相同的临时会话密钥，并使用此密钥对通信数据进行加解密。下面是对密钥协商的两个阶段具体的介绍。

4.1.1. 建立连接阶段

建立会话连接是双方能够完成正常通信的前提，若通信节点 A 要向通信节点 B 发送数据，需要在服务器端对 AB 两节点进行连接配置，服务器向节点 A 发送建立连接命令，通信节点 A 和 B 开始进行建立

连接阶段，建立连接完成后，双方向服务器反馈结果，服务器记录连接信息。图 3 为建立连接阶段流程，A 随机生成四个随机数，将四个随机数分为两组，第一组记为 CountAB、SeedAB，第二组记为 CountBA、SeedBA，CountAB、SeedAB 为通信节点 A 向 B 发送数据时计数器和密钥种子，这些密钥参数用于第二阶段的动态密钥生成，CountBA、SeedBA 是通信节点 B 向 A 发送数据时的计数器和密钥种子，两组参数相互独立，互不影响，为了防止在传输过程中因 AB 双方传输方向的不确定性，导致双方密钥参数不一致的问题，因此选择两组的密钥参数。节点 A 向节点 B 发送密钥参数时，传输方案使用上文所介绍的基于 PKI 公钥基础设施的传输方案，保证了密钥参数在传输过程中的完整性和安全性。双方将两组密钥参数保存到各自的安全存储区中，且永远不会被传出设备之外，因此除 A、B 节点其它人无法得知密钥参数。节点 B 接收到密钥参数后，将接收结果反馈给节点 A，至此第一阶段建立连接阶段完成。

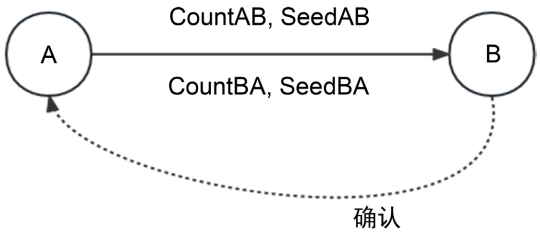

Figure 3. Process of connection establishment phase
图 3. 建立连接阶段过程

4.1.2. 密钥生成阶段

阶段二密钥生成阶段伴随着通信双方每一次的数据传输，当一方需要发送数据时需要动态生成密钥，然后使用此密钥加密数据，接收方则同样动态生成相同的密钥，使用此密钥进行解密。密钥动态生成方案基于公钥算法的密钥协商方案，提出了动态密钥生成器(Key Generator, KG)，不需要存储会话密钥，也不需要传输会话密钥，而是通过双方时刻保持一致并且不断发生变化的密钥参数，在本地动态生成会话密钥，从而保证了密钥的一致性。

节点 A 向节点 B 发送数据具体过程如图 4 所示，节点 A 端通过 KG 模块，根据 CountAB 和 SeedAB 参数生成对称会话密钥，使用此会话密钥将明文进行加密得到数据密文，再对数据明文进行 Hash 运算，最后将数据密文和明文对应的 Hash 值发送给节点 B。节点 B 收到数据后，通过 KG 模块，使用相同的密钥参数 CountAB 和 SeedAB 生成相同的对称会话密钥，并用此密钥对密文进行解密，得到明文数据，最后对比 Hash 值保证数据的完整性。若通信节点 B 向通信节点 A 发送数据，则使用另外一组密钥参数 CountBA 和 SeedBA。

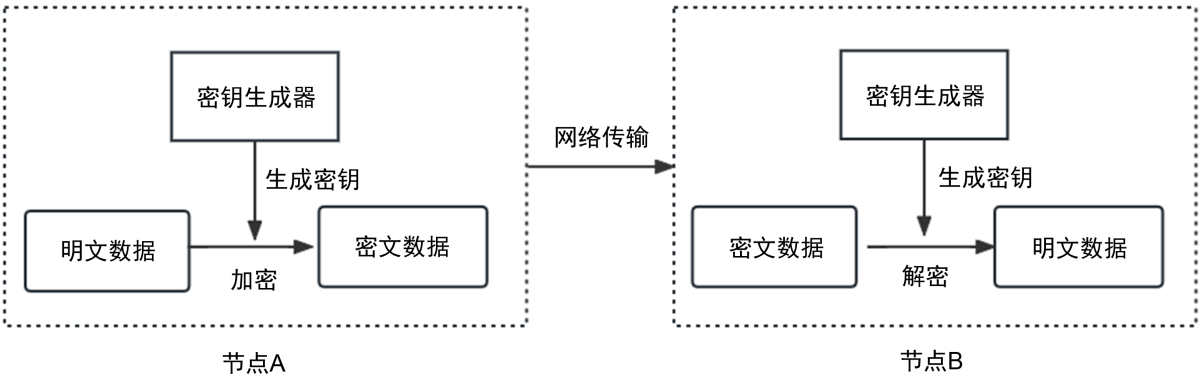

Figure 4. Process of node A sending data to node B
图 4. 节点 A 向节点 B 发送数据过程

4.2. 密钥生成器

密钥生成器是优化后密钥协商方案的核心模块，保证了数据传输双方在建立连接后不再需要交换密钥信息就可以保证会话密钥一致，并且不断变化，变化方法基于 Hash 函数实现，由于 Hash 函数的单向性，使新会话密钥无法推出旧会话密钥，因此该密钥协商方案可以支持完美前向保密特性。

下面为动态密钥生成算法具体过程：

$$\text{key1} = \text{Hash}\left(\text{CountAB} \| \text{SeedAB}\right) \tag{1}$$

$$\text{CountAB}' = \text{CountAB} + 1 \tag{2}$$

$$\text{SeesAB}' = \text{Hash}\left(\text{SeedAB}\right) \tag{3}$$

第一次 A 向 B 发送数据时，拼接 CountAB 和 SeedAB 密钥参数，并对拼接后的数字进行 Hash 运算，得到的 Hash 值作为本次数据通信的会话密钥，当数据发送成功后将更新 CountAB、SeedAB 的值，计数器 CountAB 自加 1 得到 CountAB′，SeedAB 进行 Hash 运算得到 SeedAB′。接收方 B 收到数据后同样按照公式(1)计算会话密钥，计算出会话密钥后按照公式(2)和公式(3)更新 CountAB 和 SeedAB。

4.3. 方案分析

提出的密钥协商方案设计目标是解决基于 PKI 密钥协商在支持 PFS 时性能较低的问题，以下将论述本方案如何支持 PFS 特性，以及分析了如何对密钥协商进行性能上的优化。

4.3.1. 支持完美前向保密

本方案所涉及的密钥生成算法基于 Hash 算法实现，新的会话密钥使用计数器和密钥种子作为参数，计算 Hash 值生成，Hash 算法具有单向性，无法通过 Hash 函数计算出的密文推出对应的明文，因此攻击者即使得到了本次传输时的会话密钥，也无法得到之前的密钥信息，之前的数据密文仍然无法被解密，因此方案支持了完美前向保密的特性。

4.3.2. 一次连接，多次传输

基于本方案密钥协商时节点 A 向节点 B 传输数据流程如图 5 所示，A 向 B 发送数据时，只需要建立一次连接，确认身份和同步密信息，为了保证密钥信息的安全性，需要使用基于 PKI 的密钥交换协议，因此第一阶段执行效率相对较低，性能与传统 PKI 相同，但当连接建立完成后，双方在频繁数据传输时只需要执行第二阶段，该阶段密钥生成更高效，传输时数据包负载更轻。在前文所述，基于传统 PKI 密钥协商时每一次的数据传输时都需要进行耗时的身份认证和密钥协商，而本方案的优点在于耗时的第一阶段执行频率较低，第二阶段执行频率较高，因此本方案密钥协商性能更加高效。

4.3.3. 轻量、高效地密钥协商

该密钥协商方案首先在两端建立会话连接，并且双方只发生一次的数据交换，以密钥参数的确认。在第二阶段将不需要身份认证，与基于 PKI 密钥协商相比不需要发送新的会话密钥密文和数字证书，而是直接发送密文数据和明文数据 Hash 值，该方法使用最少的交换数据量进行了数据传输，减轻了数据包的大小。并且在基于 PKI 的密钥协商方案中需要频繁的使用非对称算法加密会话密钥，接收方使用非对称密码算法进行解密得到密钥明文，非对称算法涉及耗时严重的点乘运算，时间开销较高，相较于本方案所使用的密钥生成方式 Hash 运算的方式，算法性能要慢 30 倍，效率很低，因此新密钥协商方案更加高效。

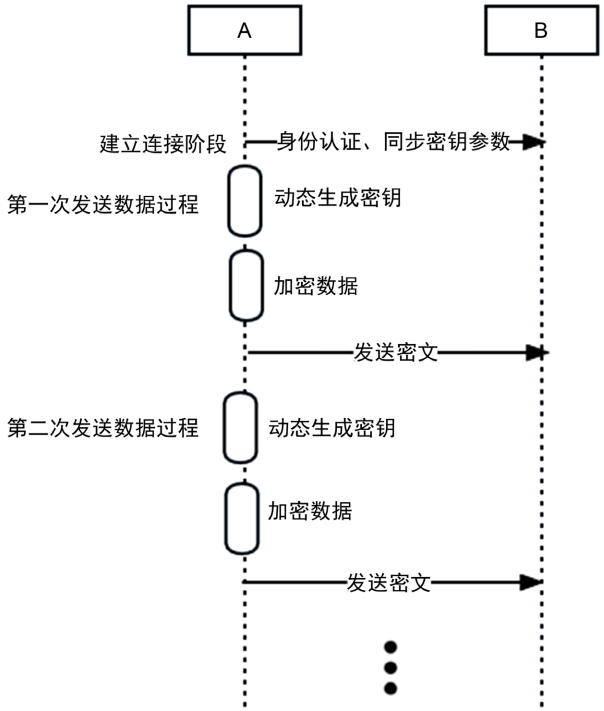

Figure 5. Data transmission process from A to B in key negotiation based on optimization scheme
图 5. 基于优化方案密钥协商时 A 向 B 传输数据过程

5. 算法实验与结果分析

实验配置

本实验使用 C++语言通过调用 GMSSL 第三方库实现提出的密钥协商方案，基于 PC 平台进行开发，开发环境如下所示：

1) CPU：Intel i7-12700；

2) 内存：16 G；

3) 操作系统：Ubuntu 18.04。

表 1 为对传统 PKI、所提出的密钥协商算法进行了运算步骤总结，由于密钥协商算法双方运算步骤是对称的，只对比了发送方在进行密钥协商时的运算步骤。传统 PKI 的密钥协商方案使用 1 次 Hash 运算 H，2 次非对称加密运算 A，3 次对称加密运算 S。本方案使用 3 次 Hash 运算，1 次对称加密运算。

Table 1. Summary of computational steps for each scheme
表 1. 各方案运算步骤总结

方案	运算
传统 PKI	H+2A+3S
优化方案	3H+ S

本节模拟了 1000 次的密钥协商并加密传输数据的过程，明文大小使用 1 KB，基于 PKI 的密钥协商各步骤消耗时间如表 2 所示，总共用时 5.580 秒，非对称算法的计算效率与其它算法相比很低，每次非对称算法只加密了很小量数据，但两次运算却占用总时间的 97%，对称算法和 Hash 算法较为接近，总占

1%，其余是系统开销时间占 2%，各运算时间占比如图 6 所示，从图中可以观察出非对称算法是密钥协商过程的瓶颈。

Table 2. Time taken for each step in PKI key negotiation process
表 2. PKI 密钥协商各步骤用时

步骤	用时
Hash 运算(1 KB)	0.011 s
SM2 签名(32 B)	2.727 s
加密明文(1 KB)	0.021 s
加密证书(2 KB)	0.025 s
加密签名(120 B)	0.003 s
SM2 加密密钥(16 B)	2.706 s
系统开销	0.087 s
总计	5.580 s

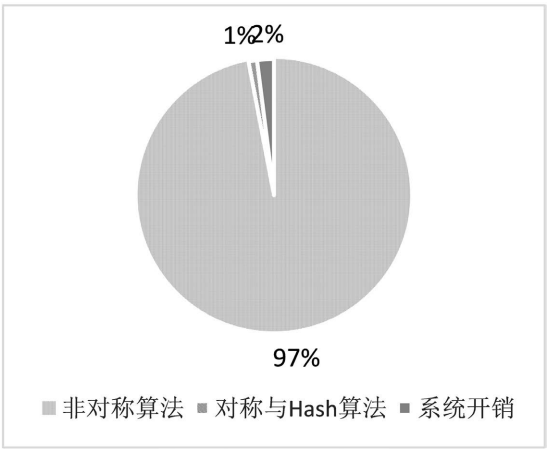

Figure 6. Proportion of time spent on various operations in PKI key negotiation
图 6. PKI 密钥协商各运算时间占比图

表 3 记录了优化后密钥协商方案的时间消耗，密钥协商了 1000 次，明文大小使用 1 KB，本方案在第二阶段传输数据替换了耗时的非对称运算，使用效率较高的 Hash 运算，在算法运行时间上大幅减少，因此打破了瓶颈，总消耗时间由 5.58 秒变为 0.122 秒，与基于 PKI 方案相比性能提升了约 55 倍。各运算时间占比如图 7 所示，其中 Hash 运算时间占比 14.7%，对称运算时间占比 13.9%，系统开销占总时间的 71%。

Table 3. Time taken for each step in the optimized key negotiation scheme
表 3. 优化后的密钥协商方案各步骤用时

步骤	用时
密钥 Hash 运算(1 KB)	0.008 s
明文 Hash 运算(1 KB)	0.009 s
加密明文(1 KB)	0.017 s

Continued

对 SeedAB 参数 Hash 运算(16 B)	0.001 s
系统开销	0.087 s
总计	0.122 s

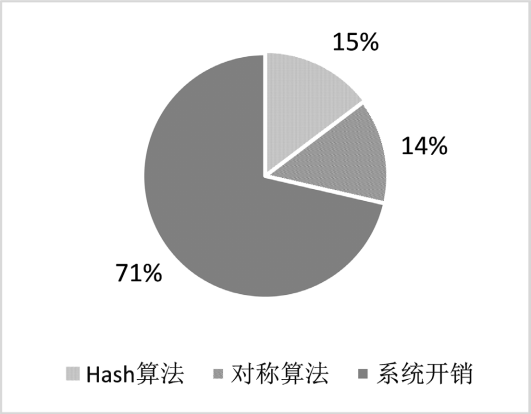

Figure 7. Percentage of operation time in optimized key negotiation scheme
图 7. 优化后密钥协商方案各运算时间占比

6. 结束语

针对基于 PKI 的密钥协商方案在支持 PFS 特性时存在大量计算开销的问题，对其性能进行了优化，优化后密钥协商方案分为两个阶段，第一个阶段建立会话连接，对通信对方进行身份认证和密钥参数的同步，第二阶段在通信节点通过密钥生成器动态生成会话密钥，保证了每次数据加密的密钥不用，并且动态变化，从而支持了 PFS 特性。对密钥协商算法性能分析，相比基于 PKI 密钥协商性能优化显著，达到了预期效果。

基金项目

国家重大研发计划——“虚拟货币关键环节风险识别与监测处置技术”(2022YFC3320900)。

参考文献

[1] Kerckhoffs, A. (1883) La Cryptographie Militaire. *Journal des Sciences Militaries*, **9**, 5.

[2] Bellare, M. and Miner, S.K. (1999) A Forward-Secure Digital Signature Scheme. In: Wiener, M., Ed., *Advances in Cryptology—CRYPTO*' 99. Springer, Berlin, Heidelberg, 431-448. https://doi.org/10.1007/3-540-48405-1_28

[3] Li, P., Su, J. and Wang, X. (2020) iTLS: Lightweight Transport-Layer Security Protocol for IoT with Minimal Latency and Perfect Forward Secrecy. *IEEE Internet of Things Journal*, **7**, 6828-6841. https://doi.org/10.1109/JIOT.2020.2988126

[4] 杨鹏飞. 工业物联网下认证密钥协商方案研究[D]: [硕士学位论文]. 西安: 长安大学, 2022.

[5] 龚成, 牛宪华, 熊玲, 等. 车载自组网中基于密钥协商的条件隐私保护认证方案[J]. 西华大学学报(自然科学版), 2022, 41(5): 73-83.

[6] 夏涛, 何俊, 刘林, 等. 无人机轻量级认证密钥协商技术研究[C]//中国指挥与控制学会. 第十届中国指挥控制大会论文集. 北京: 兵器工业出版社, 2022: 300-305.

[7] 黄晓晖, 李俊峰, 何云. 一种基于群签名密钥协商算法的多方参与完整性验证方案[J]. 信息技术与信息化, 2022(7): 102-105.

[8] 王华华, 郑明杰, 陈峰, 等. 基于 LDPC 和椭圆曲线加密算法的密钥协商方案[J]. 南京邮电大学学报(自然科学

版), 2022, 42(3): 30-35.

[9] 张萌楠. 面向物联网的轻量级安全无证书密钥管理方案研究[D]: [硕士学位论文]. 太原: 太原理工大学, 2022.

[10] 李贵勇, 张航, 韩才君, 等. 面向无线传感器网络的认证密钥协商机制[J/OL]. 小型微型计算机系统: 1-7. http://kns.cnki.net/kcms/detail/21.1106.TP.20230222.1106.014.html, 2023-03-23.

[11] Yang, Z., He, J., Tian, Y., *et al*. (2019) Faster Authenticated Key Agreement with Perfect Forward Secrecy for Industrial Internet-of-Things. *IEEE Transactions on Industrial Informatics*, **16**, 6584-6596. https://doi.org/10.1109/TII.2019.2963328

[12] Pirzada, S.J.H., Memon, Z.W., Xu, T., *et al*. (2020) Randomized Key Exchange Protocol Implementation for Internet of Things Application. 2020 *IEEE* 14*th International Conference on Open Source Systems and Technologies* (*ICOSST*), Lahore, 16-17 December 2020, 1-5.

[13] Avoine, G., Canard, S. and Ferreira, L. (2020) Symmetric-Key Authenticated Key Exchange (SAKE) with Perfect Forward Secrecy. *Topics in Cryptology—CT-RSA* 2020: *The Cryptographers' Track at the RSA Conference* 2020, San Francisco, 24-28 February 2020, 199-224. https://doi.org/10.1007/978-3-030-40186-3_10

[14] Chaeikar, S.S., Ahmadi, A., Karamizadeh, S., *et al*. (2022) SIKM—A Smart Cryptographic Key Management Framework. *Open Computer Science*, **12**, 17-26. https://doi.org/10.1515/comp-2020-0167

[15] Fan, Q., Chen, J., Shojafar, M., *et al*. (2022) SAKE*: A Symmetric Authenticated Key Exchange Protocol with Perfect Forward Secrecy for Industrial Internet of Things. *IEEE Transactions on Industrial Informatics*, **18**, 6424-6434. https://doi.org/10.1109/TII.2022.3145584