Implementação de um sistema de troca de chaves criptográficas com o auxilio de um registro de dados imutável

Pedro H. B. Lisboa[1]

***Abstract*— O presente trabalho propõe um sistema de troca de chaves públicas resistente à ataques Man-in-the-Middle. Para garantir a segurança do canal de comunicações, a implementação faz uso de um registro digital para o armazenamento das chaves públicas. Nesse artigo, é feita uma análise dos *public ledger based protocols* (PSB) apresentados em [Bui, Thanh et al. Key exchange with the help of a public ledger.]. A implementação faz uso do protocolo *Elliptic-Curve Diffie-Hellman* (ECDF) com registro centralizado e um distribuído. O desempenho das diferentes implementações são avaliadas de acordo com os critérios de desempenho e robustez contra ataques de negação de serviço.**

I. INTRODUÇÃO

O evento de troca de chaves públicas é uma etapa crucial no estabelecimento de um canal seguro de comunicações. Suponha que quando Alice deseja estabelecer uma chave compartilhada com Bob, Carol realiza um ataque MitM para interceptar a comunicação. No caso de um ataque bem sucedido, ocorrem duas trocas de chaves distintas: uma entre Alice e Carol e outra entre Carol e Bob. O sistema proposto tenta impossibilitar esse cenário ao realizar a troca de chaves por meio de um registro público de dados que seja imutável. Dessa forma, todas as chaves públicas geradas durante a troca estarão publicamente disponíveis. Ao indexá-las no registro por um contexto pré-estabelecido entre as partes, é possível detectar se o número de entradas condiz com o número de usuários legítimos e assim determinar a segurança do canal. O resto do documento está distribuído da seguinte forma: a seção II contém a fundamentação teórica dos métodos de troca de chave e dos registros imutáveis; a seção III apresenta o esquema geral do protocolo sistema; a seção IV descreve em detalhes as implementações; a seção V apresenta os resultados das simulações realizadas; a seção VI conclui o artigo.

II. FUNDAMENTOS TEÓRICOS

A. Diffie-Hellman

O sistema de chaves assimétricas surgiu como um meio de resolver o problema de distribuição de segredos para encriptação. O algoritmo proposto por Whitefield Diffie e Martin Hellman em [2] foi a primeira dessas soluções. O princípio desse tipo de troca de chaves se baseia na escolha de um modelo de geração de chaves no qual seja impossível gerar a chave privada a partir do conhecimento sobre o funcionamento do algoritmo e a chave pública. No caso do cripto-sistema de Diffie-Hellman, essa dificuldade está atrelada ao alto custo computacional para se calcular logaritmos discretos.

O algoritmo segue da seguinte forma: Duas partes comunicantes compartilham um número a e um número p que é coprimo de a e p < a. Uma das partes, Alice, gera uma chave privada X_A

e Y_A de tal forma que $Y_A = a^{X_a} \mod p$. Da mesma forma, a outra parte, Bob, gera sua chave privada X_B e pública Y_B. Note que os parâmetros a e p e a chave pública podem ser de conhecimento público de tal forma que não compromete o nível de segurança do sistema. Em seguida, as partes comunicantes trocam suas respectivas chaves públicas entre si. Com a informação recebida, Alice calcula a chave $K = Y_B^{X_a} \mod p$ e Bob chega ao mesmo resultado com sua chave privada e a chave pública de Alice. Dessa forma, ambas as partes podem fazer uso de K para encriptar suas comunicações, i.e. um canal seguro foi estabelecido.

B. Criptografia de Curva Elíptica

Diferentemente do cripto-sistema de Diffie-Hellman, esse tipo de cripto-sistema se baseia em certas propriedades de uma família de funções do tipo.

$$y^2 = x^3 + ax + b \mod p$$

Note que devido a presença do operador de módulo, todos os cálculos são feitos dentro de um campo finito. Uma das vantagens de se escolher um número primo para p é o fato de que dessa forma o campo F_p terá certas propriedades desejáveis(e.g. F_p será um fecho algébrico e todos os seus elementos serão inversíveis) para as operações algébricas utilizadas no algoritmo.[3]

A principal vantagem da criptografia de curva elíptica é que ela oferece a mesma segurança que os sistemas de Diffie-Hellman porém com um tamanho menor de chaves[2].

A descrição detalhada do funcionamento desse sistema criptográfico foge do escopo desse artigo. De forma resumida, o sistema de baseia em certas propriedades de adição e subtração de pares ordenados pertencentes à curva elítptica. Essas propriedades são tais de forma que é computacionalmente custoso a recuperação dos parâmetros originais após uma aplicação sucessiva de somas ou multiplicações. Dessa forma, o conceito dos sistemas de chaves públicas pode ser aplicado em conjunto com o sistema de curvas elípticas para se gerar um sistema de troca de chaves de Diffie Hellman em conjunto com curvas elipticas(ECDH).

*Esse trabalho não foi apoiado por nenhuma organização

[1]Pedro Lisboa é aluno de graduação no Departamento de Engenharia Eletrônica e Computação, Escola Politécnica da Universidade Federal do Rio de Janeiro, Av. Athos da Silveira Ramos, 149 - Cidade Universitária, Rio de Janeiro - RJ pedro.lisboa at land.ufrj.br

As implementações apresentadas nesse documento fazem uso desse cripto-sistema.

C. Registros públicos

Registros públicos são logs de dados que registram todo o histórico de transações desde o início de sua operação. Esse tipo de base de dados é construído de forma que a mudança de uma entrada é impossível ou muito difícil de ser realizada por um agente malicioso e que os dados inseridos possuem uma certa ordenação(geralmente temporal).

As implementações desse tipo de sistema podem ser categorizadas por dois parâmetros: centralidade do comando e publicidade dos dados. Em registros centralizados os usuários devem depositar a sua confiança em uma entidade central. Esta última pode ser responsável por qualquer ação dentro do sistema, como adicionar ou remover privilégios de leitura de geração de entradas no registro. No outro extremo do espectro, temos as implementações descentralizadas, onde a operação e integridade do sistema depende do comando dos usuários. Nesse tipo de aplicação, qualquer alteração no registro deve ser aprovada por uma maioria pré-estabelecida, o que torna o sistema robusto contra agentes maliciosos por não ter um ponto de falha(a entidade central). As redes Ethereum e Bitcoin são aplicações famosas de um registro publico descentralizado(Blockchain).

Outra forma de analisar essas estruturas é observar a publicidade dos dados. Registros privados podem restringir o acesso à uma entrada apenas para usuários pertencentes ao sistema ou até autorizar apenas que apenas um subgrupo destes tenha acesso.

III. Protocolos baseados em registros

A ideia geral do sistema discutido é impedir a ocorrência de um ataque MiTM por meio do envio e retirada das chaves públicas em um registro imutável de dados. Para simplificar o modelo, iremos considerar que nesse sistema apenas duas pessoas estão realizando uma troca de chaves no momento observado. Após o término da troca, uma das partes(pré-estabelecida pela implementação) envia as duas chaves públicas em conjunto com uma chave de contexto para um registro de dados. Após um certo tempo, ambas as partes requisitam a informação das chaves para o registro. Caso haja mais de uma entrada atrelada àquele contexto pode-se saber que o canal não é seguro e encerrar as comunicações. [1]

A. O contexto

A presença de um contexto atrelado às chaves é necessário de forma a permitir que mais de uma transação seja feita no registro. O contexto pode ser qualquer informação que seja de conhecimento das partes antes do início da transação [1], e pode depender da aplicação onde o protocolo está inserido. Uma vez que os dados registrados são imutáveis, todas as transações permanecem na cadeia de eventos de troca de chaves. Dessa forma, é necessário que o contexto contenha uma informação temporal do momento em que a troca foi realizada ou que ele seja usado somente uma vez para que assim quando a consulta ao banco for realizada, apenas uma entrada atrelada àquele contexto seja apresentada.

B. Tempo do protocolo

Para manter funcionamento do protocolo, um contexto não pode ser usado mais de uma vez. Para aplicações onde o estabelecimento de um novo contexto não pode ser realizado toda vez que duas partes estabelecerem um canal seguro, pode-se dar um tempo de vida para aquela transação ou até inserir uma informação sobre o tempo na própria chave de contexto.

C. Aplicações

Algumas aplicações que podem fazer uso desse tipo de troca de chave incluem sistemas de chat peer-to-peer e sistemas de compartilhamento de arquivos via bluetooth ou wi-fi. [1] No primeiro, a chave de contexto pode ser estabelecida a partir dos identificadores de usuários ou,se a aplicação suportar VoIP, os números de telefone das partes. Para transferência de arquivos em dispositivos móveis, muitas das vezes os usuários estarão fisicamente próximos e poderão chegar à um acordo sobre uma chave de contexto que não necessariamente precise envolver o tempo da transação. Para esse caso, deve-se avaliar também a possibilidade de limpeza do banco de dados caso o registro seja centralizado, ou criação de uma nova *blockchain* de forma periódica: como o contexto é uma chave criada pelos usuários e pode ser usada apenas uma vez, as possibilidades de chave diminuirão de acordo com o tempo em que o sistema estiver em serviço.

IV. Implementação

Para testar a implementação desse protocolo, duas implementações foram feitas. A diferença entre elas se encontra no tipo de registro imutável utilizado e suas propriedades. O sistema de geração de chaves Diffie-Hellman utilizado foi o Curve22519, um cripto-sistema baseado em curvas elípticas. Uma das vantagens dessa implementação está em seu baixo custo computacional e o fato de não ter sido patenteado. [6] O contexto utilizado foi formado a partir de strings identificadores de cada usuário.

A. Registro centralizado

Para a simulação, foi utilizado o Banco de dados Não Relacional MongoDB como forma de simular o controle e uma entidade central sobre os dados. O sistema foi construído de forma a atribuir um usuário e senha para cada uma das partes e desabilitar a capacidade destes de remover uma entrada do banco. O sistema foi pensado de forma que a única forma visualização é por meio de uma conta de usuário, i.e. o registro é privado.

B. Registro distribuído

A geração da blockchain foi feita através da plataforma Multichain, que permite um alto nível de configuração dos parâmetros, tanto da cadeia em sí como dos usuários. Como o único uso da blockchain necessário para o funcionamento do protocolo é o registro e retirada de dados, boa parte das funcionalidades da plataforma não foi necessária. Para

estabelecer uma funcionalidade básica para a realização das simulações, foi garantido os usuários a permissão da criação de streams de dados dentro da cadeia.[7]

Registros públicos e privados: As configurações da plataforma permitem que um um registro criado possa ser tanto público quanto privado. As simulações foram realizadas em uma implementação cujo acesso à leitura era restrito aos membros da blockchain (i.e. os usuários da aplicação).

C. Pseudo-código geral

Algorithm 1 Protocolo ECDF com Registro imutável

Input: identificação do par de comunicação
Output: chave simétrica compartilhada

```
Determinar tempo máximo da troca
Entrar com as credenciais no registro
Gerar chave privada X
Gerar chave pública Y
Enviar Y para a outra parte
Receber a chave pública da outra parte
if (Usuário que iniciou a comunicação) then
    Enviar o registro com o contexto pré-estabelecido
end if
Consultar Registro dado o contexto
if Mais do que uma entrada com o contexto then
    Terminar comunicações
end if
Calcular chave compartilhada K
return K
```

D. O fator de tempo limite

O estabelecimento de um tempo limite para a transação deve ser feito pensando na aplicação na qual a implementação está inserida. As simulações na seguinte seção fazem uma análise do tempo tomado para a realização das operações, mas esse estudo deve ser realizado também para cada caso.

E. Suscetibilidade à ataques de negação de serviço

O sistema de autenticação utilizado no registro central garante que apenas usuários registrados no banco de dados possam gerar entradas de forma a prejudicar a troca de chaves dos usuários. Em aplicações como a transferência de dados via *bluetooth*, onde a escolha do contexto é escolhida na hora pelos usuários, a necessidade do atacante precisar adivinhar o contexto para o envio das chaves consiste em uma camada extra de proteção contra esses ataques. Dessa forma, ataques de negação de serviço que buscam explorar alguma vulnerabilidade intrínseca na escolha do contexto podem ser evitados com um sistema de autenticação e controle do próprio registro. Vale ressaltar que a plataforma utilizada para implementar o registro descentralizado possibilita tanto a configuração das permissões dos usuários quanto a edição de parâmetros da própria *blockchain*, possibilitando o combate à ataques DoS de forma semelhante ao caso com o registro centralizado. [1]

F. Ataques de falsificação do usuário

O objetivo de uma interceptação de mensagens via MitM pode não ser a possibilidade de decriptar as mensagens entre as partes comunicantes mas se passar por uma delas. Caso um ataque desse gênero seja feito, o protocolo apresentado não oferece nenhum mecanismo para vítima detectar a falsificação. O atacante, ao não repassar as mensagens para um dos usuários e realizar a comunicação direta com a vítima simulando ser a outra parte comunicante, garante que apenas uma entrada seja enviada para o registro. Resta apenas que o contexto seja forjado de forma que a vítima não perceba que o canal está comprometido. A forja da chave de contexto depende diretamente da aplicação fim na qual o protocolo foi implementado. Um caso em que a implementação seria resistente à esse tipo de ataque é um sistema de comunicação via VoIP. Mesmo que o contexto seja forjado(e.g. o atacante, sabendo que o contexto é formado pela concatenação dos números de telefone das partes, se apresenta com o número da vítima a qual ele está personificando) podemos assumir que, na maioria das vezes, as partes comunicantes são capazes de reconhecer o tom de voz do indivíduo com quem esta falando por já se conhecerem pessoalmente.

Na aplicação de troca de arquivos em sistemas móveis, a robustez do sistema contra ataques de falsificação é ainda maior: os usuários envolvidos na transação estarão fisicamente próximos e podem combinar uma chave de contexto forte o suficiente de forma que esta não possa ser adivinhada pelo atacante durante a transação. Ainda sim, nesse caso a segurança do sistema estaria dependente da capacidade dos usuários em gerar um contexto resistente à ataques de dicionário ou força bruta. Seria então recomendável que a aplicação garantisse que o contexto criado tenha um nível mínimo de segurança (e.g. forçando que a chave tenha no mínimo um certo comprimento e contenha caracteres especiais e alfanuméricos).

V. Simulações e Resultados

A. Configuração do hardware

Os testes foram realizados em três computadores com diferentes configurações dentro de uma rede local. O ataque MiTM foi feito a partir de uma máquina com processador Intel I7 de 5ª geração, 4 Gigabytes de memória RAM, rodando um sistema Ubuntu 16.04LTR de 64 bits. Um dos pontos de comunicação foi implementado em um Intel Core I3 de 4ª geração, 4 Gigabytes de RAM, rodando um sistema Windows 10 de 64 bits. A ultima máquina, na qual o outro ponto foi implementado e foram coletados os dados para a análise do tempo de execução, possui um Processador Pentium Dual-Core, 2 Gigabytes de memória RAM e roda um sistema Windows 7 de 32 bits.

B. Tentativa de ataque MitM

Um ataque MitM foi realizado como *baseline* para a aplicação do algoritmo. O ataque foi realizado utilizando a ferramenta ettercap para a realização de um ARP Poisoning e a visualização dos pacotes feitas por meio do Wireshark. Em seguida, um novo ataque foi aplicado contra o sistema

discutido nesse artigo. Nesse caso, ambos os usuários foram capazes de detectar a presença de uma entrada a mais no banco de dados.

C. Análise de desempenho

Para cada um dos casos, foram rodadas 400 simulações para a estimação da média de operação das trocas. As medições foram realizadas na máquina e, a partir da média e variância amostral, o valor para a média real do processo foi estimado por meio de um teste de hipótese com a distribuição t de Student. As simulações foram realizadas durante um período de 3 dias não consecutivos.

TABELA I
INTERVALO DE CONFIANÇA PARA A MÉDIA DO TEMPO DE EXECUÇÃO COM 95% DE SIGNIFICÂNCIA

	Min(segundos)	Max(segundos)
Sem registro	1.6913	1.8133
Registro centralizado	5.332523	5.386523
Registro descentralizado	5.318391	5.37439

VI. Conclusões e direções futuras

O uso do protocolo baseado em registros imutáveis se mostrou uma boa solução para mitigar ataques MitM em sistemas de comunicações e transmissão de arquivos. A variação da centralidade do registro não influenciou muito no desempenho do algoritmo, apontando que os valores de tempo limite para a troca de chaves podem ser semelhantes para esses dois casos. As aplicações que fazem uso de um registro privado possuem maior robustez contra ataques de negação de serviço que possam tentar explorar propriedades intrínsecas do protocolo descrito. Futuras pesquisas nesse tema podem buscar desenvolver uma plataforma funcional de chat peer-to-peer ou troca de arquivos via bluetooth com um esquema de troca de chaves baseado na implementação apresentada. Outra direção que pode ser tomada é na realização de novas análises do desempenho da troca de chaves em ambientes e configurações de rede diferentes dos que foram considerados nesse artigo.

Agradecimentos

Gostaria de agradecer aos Professores Luis Felipe de Moraes e Evandro Macedo, pela oportunidade de fazer este trabalho e aos conhecimentos compartilhados durante as aulas de Tópicos Especiais em Sistemas de Comunicação. Gostaria também de agradecer ao Eduardo Alves, que disponibilizou o hardware necessário para a realização do experimento.

References

[1] Bui, Thanh, Aura, Tuomas. (2017). Key exchange with the help of a public ledger. Cambridge International Workshop on Security Protocols 2017
[2] Diffie, Hellman. New Directions in Cryptography. 644 IEEE Transactions On Information Theory, Vol. It-22, No. 6, novembro de 1976.
[3] ECC. https://www.johannes-bauer.com/compsci/ecc/ [Novembro, 2017]
[4] Standards for Efficient Cryptography Group (SECG), SEC 1: Elliptic Curve Cryptography, Version 1.0, September 20, 2000.
[5] B. Aziz and G. Hamilton, "Detecting Man-in-the-Middle Attacks by Precise Timing,"2009 Third International Conference on Emerging Security Information, Systems and Technologies, Athens, Glyfada, 2009, pp. 81-86.
[6] Bernsteinm, D. Irrelevant patents on elliptic-curve cryptography. https://cr.yp.to/ecdh/patents.html
[7] Multichain. https://www.multichain.com/download/MultiChain-White-Paper.pdf