Неотказуемость, цифровая подпись и хеш в криптографии

Хэш - это короткая строка текста, которую мы можем создать на основе данных, содержащихся в открытом тексте. Иногда это называют дайджестом сообщения или чем-то вроде отпечатка пальца. Значит, если что-то изменится с этими данными, у нас будет другой отпечаток пальца или другой хэш. Это то же самое, что настоящий отпечаток пальца. Если человек изменится, то увидите, что отпечаток очень разный.

Хотя хеш очень хорошо проверяет целостность данных, он не связывает эти данные с конкретным человеком. Мы можем убедиться, что данные, которые мы получили, точно такие же, как и данные, которые были отправлены. Но мы не можем проверить, кто прислал данные. Однако есть способы обеспечить эту дополнительную целостность.

Посмотрим, как работает это хеширование, на практическом примере. В Интернете есть организация Project Gutenberg, которая опубликовала Энциклопедию Гутенберга. Я скачал первый том той энциклопедии, а это 8,1 мегабайта данных. А затем я запустил приложение, которое взяло все эти данные и создало хэш или отпечаток первого тома энциклопедии.

Теперь, если бы мне пришлось изменить один символ внутри этого файла, где угодно, хотя размер файла после внесения изменения точно такой же, где-то в этом стоге данных есть какая-то разница. Но человеку было бы очень трудно прочитать все эти данные, 8,1 мегабайта текста, и каким-то образом определить, где могут быть эти отдельные изменения.

Но если выполнить хеш изменённых данных, то вы увидите, что значение хеша, которое я создаю, сильно отличается от значения хеша, которое было оригинальным. Итак, если я загрузил этот файл, выполнил свой собственный хэш и сравнил его с оригиналом, я вижу, что что-то действительно изменилось с этим конкретным томом одной из Энциклопедии Гутенберга.

Используя эти хэши, мы можем предоставить доказательство целостности. Мы знаем, было ли что-то изменено, когда информация была отправлена от первоначального отправителя.

Но мы также можем добавить к этому дополнительный уровень целостности, **называемый доказательством происхождения,** где мы можем проверить человека, который отправил нам данные. Иногда вы увидите, что это называется аутентификацией, когда мы смотрим на источник сообщения. **Используя цифровую подпись**, мы обеспечиваем неотказуемость. Таким образом, мы не только знаем человека, который отправил нам эти данные, но и любой другой мог изучить эту транзакцию и убедиться, что полученная нами информация действительно исходила от отправляющей стороны.

Точно так же, как кто-то может использовать ручку и бумагу, чтобы подписать контракт и отправить его вам, в криптографии мы используем цифровую подпись. Эта цифровая подпись использует закрытый ключ, который известен только тому, кто отправляет данные. Больше ни у кого нет копии этого закрытого ключа. Чтобы убедиться в использовании закрытого ключа, мы используем открытый ключ, связанный с этим закрытым ключом. И таким образом, мы можем заверить, что полученная нами информация не только такая же, как отправленная, но мы знаем, что ее должен был отправить человек, предоставивший цифровую подпись.

На практике добавление цифровой подписи к документу обычно создается путем нажатия на поле с надписью "Добавить цифровую подпись". И много криптографии происходит за кулисами. Давайте немного поднимем капот и посмотрим, какой может быть этот процесс, когда вы установите этот флажок, чтобы добавить цифровую подпись.

Начнем с разговора, который происходит между Алисой и Бобом. Алиса отправляет Бобу сообщение, в котором говорится: “Вы наняты, Bob.” И первое, что Алиса сделает, это предоставит цифровую подпись, прежде чем она ее отправит. Первое, что происходит, когда она нажимает этот флажок для цифровой подписи, это то, что алгоритм хеширования создает хэш этого открытого текста. В данном случае открытый текст “Вы наняты, Bob.”

Как только этот хэш будет создан, нам теперь понадобится какой-то способ убедиться, что он действительно исходил от Алисы. А поскольку Алиса единственная со своим закрытым ключом, мы собираемся зашифровать этот хэш закрытым ключом Алисы, взять этот зашифрованный хэш, отправить его вместе с открытым текстом, чтобы Боб получил сообщение, в котором говорится: “Вы наняты, Боб.” И к этому сообщению прикреплена цифровая подпись.

В большинстве случаев Алиса собирается отправить это сообщение Бобу, используя электронную почту или какой-либо другой тип электронной доставки. Боб собирается получить именно то сообщение, которое было отправлено, которое говорит “Вы наняты, Bob” с цифровой

подписью, прилагаемой к нему. Боб собирается использовать открытый ключ Алисы, который является ключом, доступным любому, и он собирается изучить цифровую подпись и расшифровать ее с помощью этого открытого ключа.

Как только это расшифровка произойдет, у нас появится исходный хэш, созданный для этого открытого текстового сообщения. И в этот момент Боб хочет посмотреть, соответствует ли хэш, который был в этой цифровой подписи, хэшу того, что он получил. Так что он собирается выполнять ту же функцию хеширования, которую изначально выполняла Алиса. Боб собирается взять исходный открытый текст и запустить его через тот же алгоритм хеширования, чтобы придумать хэш того, что он получил.

Теперь Боб может провести сравнение, чтобы увидеть, совпадает ли хэш, включенный в цифровую подпись, с хэшем, который он создал вручную из открытого текста. И если это совпадает, мы не только знаем, что полученная нами информация точно такая же, как и информация, которая была отправлена, но мы знаем, что ее нужно было отправить от Алисы.