SECUREDROP

Deniability primer

Giulio

Deniability #1 - Basic types

- There is not really a single and standard way to define it, but let's try:
  a. **Deniable encryption**

     True/VeraCrypt style. It is impossible to prove a certain encrypted volume or message exists. In the case of True/Veracrypt, it is somewhat doubled with steganography: we have a decoy volume and an hidden one.

  b. **Deniable authentication**

     Applies to messaging protocols, OTRv2, x3DH: we need to guarantee the authenticity of messages, without implying cryptographic undeniable proofs that the sender sent it.

Deniability #2

- Case (b) is what we need for SD 2.0. We cannot deny SD 2.0 is an encrypted messaging system, or hide on the server side that messages exists at all. We can of course add decoy traffic and that should remain indistinguishable.
- Authentication deniability usually require at least a party to be compromised or willingly provide transcripts to be useful: we assume access to key exchanges and plaintext materials.

Deniability #3 - Some kind of judge

2. DENIABILITY

When we discuss deniability, we must do so with respect to an action and a type of judge. We say that an action is deniable with respect to a given judge if the judge cannot be convinced that an individual performed the action. To make such a statement, we need to define the environment in which the judge resides, and the type of evidence that is required to convince the judge that the action was performed. If an action is deniable with respect to a judge, we say that individuals can "plausibly deny" performing the action. Note that this deniability does not constitute a proof that the parties *did not* perform the action; plausible deniability simply denotes a lack of convincing proof.

Of course, (in literature) a judge can only be convinced by cryptographic proofs!

Deniability #4 - Message VS Participation

There are two primary aspects of conversations that can be called deniable. We can say that messages transmitted during a conversation are deniable (*message repudiation*), but we can also say that participation in the conversation itself is deniable (*participation repudiation*). These properties are orthogonal; a protocol may offer one or the other, both, or neither. For example, messages sent using the well-known OpenPGP protocol are signed with the sender's long-term key, but the signed message does not include the recipient's identity. An OpenPGP-signed email can be used as proof that the message was signed, and presumably authored, by the sender, but not that the sender was in a conversation with the ostensible recipient. Consequently, OpenPGP offers participation repudiation but not message repudiation.

Deniability #5 - Disclaimer

that an action is deniable. In the secure messaging literature, it is common to consider only judges that are completely rational, and decide on the plausibility of an event based solely on the evidence presented to them. The only acceptable evidence for these judges is a valid cryptographic proof, verifiable by the judge, showing that the event must have occurred. In reality, of course, judges are more lenient, and routinely accept plaintext transcripts as evidence. The goal of deniable protocols is to not supply *additional* evidence against a participant, in the form of a hard-to-deny cryptographic proof. Concretely, a messaging protocol that digitally signs every message with the sender's long-term key would not satisfy our notion of deniability, while an unencrypted and unauthenticated protocol would.

X3DH - Deniable Key Exchange (DKE)

- Implicit authentication
- Message repudiation
- Forward secrecy

3.1. Overview

X3DH has three phases:

1. Bob publishes his identity key and prekeys to a server.
2. Alice fetches a "prekey bundle" from the server, and uses it to send an initial message to Bob.
3. Bob receives and processes Alice's initial message.

The following sections explain these phases.

3.2. Publishing keys

Bob publishes a set of elliptic curve public keys to the server, containing:

- Bob's identity key IK_B
- Bob's signed prekey SPK_B
- Bob's prekey signature $Sig(IK_B, Encode(SPK_B))$
- A set of Bob's one-time prekeys (OPK_B^1, OPK_B^2, OPK_B^3, ...)

X3DH - Shared key

If the bundle does not contain a one-time prekey, she calculates:

$DH1 = DH(IK_A, SPK_B)$

$DH2 = DH(EK_A, IK_B)$

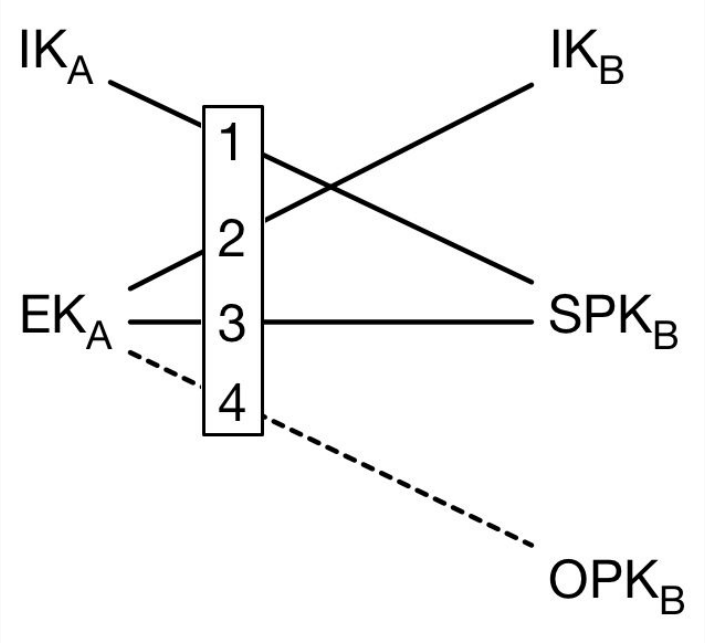
$DH3 = DH(EK_A, SPK_B)$

$SK = KDF(DH1 \,||\, DH2 \,||\, DH3)$

If the bundle *does* contain a one-time prekey, the calculation is modified to include an additional *DH*:

$DH4 = DH(EK_A, OPK_B)$

$SK = KDF(DH1 \,||\, DH2 \,||\, DH3 \,||\, DH4)$



X3DH - Why is it implicitly authenticated?

- Usage of long-term identity keys in the shared key calculation
- Only Bob or Alice can compute that shared key
- Every party know if they are honest or not
- So the message can only be forged from the other party

X3DH - Why is it deniable?

- Deniable as in: Bob can present a cryptographic transcript of Alice's message from his phone to an offline judge and that is not undeniable crypto proof.
- The judge does not know if Bob is honest or not
- So the messages can have been forged by EITHER Bob OR Alice
- Because there is no signatures, and the only "evidence" is the shared key usage

X3DH - Requirements - Notes

- **Both Alice and Bob identities must be advertised** - they can be fetched from the server knowing a key (the phone number or an id)
- **The communication itself on the server is non-deniable** - a message from Alice ends up in Bob's delivery queue
- **Sealed sender does not apply during the key-exchange**

Alice then sends Bob an initial message containing:

- Alice's identity key IK_A
- Alice's ephemeral key EK_A
- Identifiers stating which of Bob's prekeys Alice used
- An initial ciphertext encrypted with some AEAD encryption scheme [4] using *AD* as associated data and using an encryption key which is either *SK* or the output from some cryptographic PRF keyed by *SK*.

SDNG - Deniability

Multiple meanings:

- Local Deniability: no state/persistence on whistleblower machine
- Server User Deniability: no accounts on server (no user-enumeration)
- Message Repudiation: possible Signal like deniability on the conversation?

SDNG - We can’t do X3DH

- Keys needs to be public and referenced
- We want everything to be hidden from the server
- A source does not advertise keys
- Can the first contact be deniable? -> Most likely no
- What can we do after the first contact?
  - Maybe x3dh?
  - Maybe leak the source intermediate secrets?
  - Or directly send the journalists the source passphrase?

Real world?

- Screenshots are accepted 99% of the times
- Only when forging is shown easy deniability holds
- Crypto deniability does not imply logical deniability
- Forging tools should be built into the applications
- Suspect Signal does not provide the tools on purpose

Questions?

References

- 2011, One-round Strongly Secure Key Exchange with Perfect Forward Secrecy and Deniability
- 2015, Deniable Key Exchanges for Secure Messaging
- 2016, The X3DH Key Agreement Protocol