Dissent: Accountable Anonymous Group Messaging
Erratum

Henry Corrigan-Gibbs
henry.corrigan-gibbs@aya.yale.edu

Bryan Ford
bryan.ford@yale.edu

Introduction

Dissent [2] is a protocol for sender-anonymous group-wise broadcast, which builds on an anonymous data collection protocol by Brickell and Shmatikov [1]. Since publication of Dissent, we have discovered a replay attack against both Dissent's fixed-length shuffle protocol and the original Brickell-Shmatikov protocol. Herein we describe this attack and outline its implications. The attack may be prevented by requiring group members to generate fresh primary encryption/decryption keypairs before every protocol round, or by prepending a per-session nonce to expose replayed ciphertexts. We present the following attack as a representative member of a larger set of similar attacks, all of which can be prevented with the same means.

Replay Attack

This is an attack against both the Brickell-Shmatikov anonymous data collection protocol [1] and Dissent's [2] fixed-length shuffle.

After at least one protocol round has been successfully completed, an adversary can misbehave in subsequent protocol rounds so as to break the anonymity of the messages sent in the first (successfully completed) round. This particular form of the attack assumes that the adversary controls the first group member in the ordering of participants. Mounting the attack further requires that: (1) all participants reuse their primary (long-term) encryption keypairs over many protocol rounds, and (2) the attacker participate in at least two such protocol rounds.

Attack Summary Let the set of participating group members in a round of the Brickell-Shmatikov anonymous data collection protocol or Dissent's fixed-length shuffle be $P = \{p_1, p_2, \ldots, p_N\}$. We assume that p_1 is an adversarial group member.

The attackers begin by participating honestly in a round r of either protocol that terminates in the Decryption phase. In round r, all participants learn the correspondence of secondary ciphertexts from round r, $\{C_1'^r, \ldots, C_N'^r\}$, with the plaintext messages, $\{m_1^r, \ldots, m_N^r\}$, from round r. Later on, the attackers participate in an "attack" round a of the protocol with the same group of participants. In the attack round, the dishonest first group member p_1 does not shuffle the true primary ciphertext list $(C_1^a, \ldots, C_N^a)$, but instead shuffles a specially crafted ciphertext list:

$$(v_1, \ldots, v_{N-1}, C_t^r)$$

Here, $\{v_1, \ldots, v_{N-1}\}$ are random values drawn from the message space and serially encrypted with all group members' secondary public keys *from round* r and all group members' long-term primary public keys. C_t^r is the primary ciphertext of the target group member p_t, whose anonymity the adversary wants to break.

After p_1 performs the shuffle-and-decrypt operation with the dishonest ciphertext set, all other group members continue to perform the Anonymization phase honestly. Group members then proceed to the Verification phase, where p_N publishes a permutation of secondary ciphertext list:

$$(C_1^a, \ldots, C_N^a)$$

Since p_1 substituted the ciphertexts at the start of the Anonymization phase, the secondary ciphertext list is equal to some permutation of the list:

$$(D(v_1), \ldots, D(v_{N-1}), C_t^{\prime r})$$

where $D(v)$ denotes the decryption of v with all participants' primary private keys.

Since the adversary selected the random message values herself and performed the public-key encryptions to produce $\{v_1, \ldots, v_{N-1}\}$, she knows their decryptions $\{D(v_1), \ldots, D(v_{N-1})\}$. The attacker identifies the value $C_t^{\prime r}$ as the *only* secondary ciphertext in the Verification phase of the attack round that she does not know.

Using this technique, the attacker learns which primary ciphertext C_t^r decrypts to the secondary ciphertext $C_t^{\prime r}$. The attacker already knows, from round r, the decryption of $C_t^{\prime r}$. Combining these two pieces of information, the attacker learns the plaintext of target t's anonymous message and breaks the anonymity of target t.

Implications for Brickell-Shmatikov This attack completely breaks the anonymity of the Brickell-Shmatikov protocol over multiple protocol rounds. Since the Brickell-Shmatikov protocol has no means by which honest participants can identify attackers, group member p_1 can perform this attack many times with a different target. Acting unilaterally, p_1 can learn the plaintext of every honest participant from a previous protocol round after running one attack round per victim.

Implications for Dissent Dissent's accountability property ensures that honest group members will expose one faulty group member, if one exists, at the conclusion of every protocol round. Since p_1 in this case is the only misbehaving group member in the attack scenario, p_1 will be the only group member exposed at the end of an attack round of the protocol.

If the adversary controls the first $\lceil \frac{N}{2} \rceil$ members of the group, then the adversary can run $\lceil \frac{N}{2} \rceil$ attack rounds, matching one honest participant to her plaintext in each attack round. In this fashion, an adversary who controls one attacking group member per honest group can break the anonymity of every group member for a given round in $\lceil \frac{N}{2} \rceil$ subsequent protocol rounds.

Prevention

Group members can prevent these attacks by generating new primary encryption keypairs for every round of the protocol. Alternatively, group members could prepend a per-round nonce to every message before encryption and only publish decrypted messages that begin with the correct nonce.

References

[1] Justin Brickell and Vitaly Shmatikov. Efficient anonymity-preserving data collection. In *12th KDD*, August 2006.

[2] Henry Corrigan-Gibbs and Bryan Ford. Dissent: accountable anonymous group messaging. In *CCS*, pages 340–350, October 2010.