Broadcast Behavior in a Closed System

Evans, Ben P. `ben.evans@rutgers.edu`

Quinn, Maloy S. `mquinn@plinthist.com`

Abstract

We present a retroactively insecure cryptographic protocol, which implements verification matching the behavior of broadcast-only protocols (e.g. radio, IRC[1]) in a system where a complete but unreliable record of transmission exists. By intentionally degrading the security of cryptographic identities used in previous cycles, we're able to achieve cryptograhic broadcast behavior in systems where every packet is monitored and logged.

Contents

1 Description

By degrading the security of past gates, we are able to achieve the intended functionality. We implement a 'buffer' of cryptographically 'true' packets, which are the only provably valid packets for their respective cryptographic identities. As the 'head' of the transmission extends, these cryptographic identities are intentionally leaked. This removes the provability of any packets with the same identity, thereby rendering them indistinguishable from equivalently signed packets with altered contents to any user without a reliable record of the transmission's history.

2 Generalized Structure

The following model describes the full range of transmissions that apply a public-key cryptographic system to achieve non-static verifiability. The unit of data broadcasting in this paper is the node. In addition to arbitrary content, each node is allowed to carry a set of public keys and a set of private keys. These metadata have the ability to alter the potential for 'trust' in other nodes – the verifiability of their connection to the greater transmission. By using public keys introduced on a trusted node, receivers can extend that trust to the nodes those keys sign. This process, however, is only possible prior to the release of the corresponding private keys – once this trivializes their usage in signature, establishing new trust with these key-pairs becomes impossible.

2.1 Representing a Transmission

The information of a transmission with these features can be expressed as a pairing of a set of nodes N with a set of key-pairs K, a total order $\leq$ on N encoding chronology, and a triplet of binary relations I, S, and R between the two sets. These represent introduction (that a node publishes a key-pair's public key), signature (that a node is signed with a key-pair), and release (that a node publishes a key-pair's private key), respectively.

In principle, any such structure corresponds to a transmission – if no particular intentions are assumed, a transmitter may choose to publish any information at any time. However, without compromising the complete depiction of cryptographic broadcasting, we can exclude structures that publish information which is incapable of affecting interpretation. This, the presence of a coherent broadcast order, here is defined as the absence of binary relations which are incapable of affecting trust at the time of publication. There are two major classes of incoherency:

1. The effect of a release on trust is unrelated to the node which carries it except in chronology. Hence, releasing a given key-pair more than once is equivalent to only releasing it in the earliest instance.

2. Once a key-pair is released, signature through that key-pair is entirely falsifiable and thus meaningless. Since the only effect of introduction is to allow signature to confer trust, this operation is likewise impotent beyond that point.

Thus, a structure $(N, K, \leq, I, S, R)$ corresponds to a coherent transmission if and only if stipulations (1) and (2) hold:

$$\text{For all } n_a, n_b \in N, k \in K$$
$$(n_a Rk \wedge n_b Rk) \implies n_a = n_b \tag{1}$$

$$\text{For all } n_a, n_b \in N, k \in K,$$
$$[(n_a Rk) \wedge (n_b Sk \vee n_b Ik)] \implies n_a > n_b \tag{2}$$

These maintain the uniquity of release (1) and the strict succedence of release relative to introduction and signature (2).

2.2 Receiver

$T(i) \subseteq N$ is the set of nodes trusted by a given receiver immediately after node i is published. It's assumed that T is fixed by the statement $B \subseteq T(b)$ for some finite ordinal b and set of nodes B, with T being defined only for $i \geq b$. For any finite ordinal i on which T is defined, $T(i+1)$ contains a node $n_{x \leq i+1}$ if and only if it meets at least one of the conditions (3) and (4):

$$n_x \in T(i) \tag{3}$$

$$\text{For some } n_y \in T(i+1), k \in K, z \leq i+1$$
$$n_y Ik \wedge n_x Sk \wedge [\forall n_z \neg (n_z Rk)] \tag{4}$$

That is, trust in nodes persists on the assumption that trusted nodes are remembered as such, and new trust can be established when a trusted node introduces another node through an unreleased key-pair.

2.3 Describing Transmissions

The model of transmission presented above is sufficient for both complete and incomplete instances. However, there is the clear potential for redundancy, and its significance varies based on the intention of continuation.

In complete transmissions, any key-pair which has no effect on trust – which is irrelevant to T – can be labeled *eliminable*, and a transmission without any such key-pairs can be described as *key-pair-minimal*. Note, however, that eliminable key-pairs in an incomplete transmission can be used by a continuation to affect trust.

Still, we observe that a key-pair cannot affect trust beyond the point at which it is released. If $n_i Rk$, condition (4) can never be satisfied by k at or beyond $T(i)$. Consequently, if a key which is at some point released can be removed from a structure without affecting trust, no continuation allows this key-pair to affect trust. Such key-pairs can be termed *eliminable by release*, and a transmission without any is *key-pair-minimal by release*.

3 Implementation

The actual transmission of an SWN structure follows straightforwardly from its construction. Each node can be encoded as a packet containing its contents and the keys that it has used to publish. The only constraint on the structure of packets is that the set of such keys for each of I, S, and R can be reconstructed and the associated data (key or signature) of each extracted, along with the node's content. As previously mentioned, whenever an introduction or signature occures with or after a release for the same key-pair, it has no effect, since the ability to create messages verifiable through said key-pair is trivially available depending on the intended outcome. A graphical form follows simply from the SWN structure; we thus present a linearly constrained form to demonstrate the malleable nature of the SWN structure.

3.1 Linear Form

Suppose we have a broadcast in which there is a total intended reception order of packets, matching the order of their publication. We can then consider the set of SWN structures where introduction and release follow this intended order in a linear fashion.

3.1.1 The Marginal Section

At any given time-step, this constraint will manifest as a continuity of those packets which are both introduced and unreleased. Any such packet must be ahead of all released packets, and behind all unintroduced packets, meaning that any packet between two such packets must be likewise. Thus, the set of these packets is an unbroken series. We can describe the shape of this series, referred to hereafter as the marginal section, in terms of the backward and forward margins $M_b(\alpha)$ and $M_f(\alpha)$ immediately after packet α is published. The former includes marginal packets which precede α, and the latter those which succeed it.

3.1.2 Changes in Margins

The progression of these margins can then be tracked based on the relations formed by each packet. Since the marginal section must always remain continuous, each packet can only a continuous series of packets from its lower end, and can only introduce a series extending the upper end. The SWN behavior of each linear packet can thus be described purely in terms of the number of packets released and introduced, respectively, and equivalently in terms of modifying the marginal section. These can be denoted $\Delta_b(\alpha)$, for release, and $\Delta_f(\alpha)$, for introduction.

Relative to α's predecessor, these backward and forward Δ-values determine (respectively) the change in the length of the backward and forward margins from α to $\alpha + 1$. The successive values of $|M_b|$ and $|M - f|$ can then be traced in terms of per-packet introduction and release as follows:

$$|M_b(0)| = 0 \tag{5}$$

$$|M_f(0)| = \Delta_f(0) \tag{6}$$

$$|M_b(\alpha+1)| = |M_b(\alpha)| - \Delta_b(\alpha) + 1 \tag{7}$$

$$|M_f(\alpha+1)| = |M_f(\alpha)| + \Delta_f(\alpha) - 1 \tag{8}$$

This system takes into account the shift of the focused packet; from α to $\alpha+1$, M_b gains packet α, while M_f loses the same. The transmission is over when $M_f(\alpha)$ is empty; all introduced packets must be $\leq \alpha$ and hence have been published.

According to these relationships, a broadcaster can select Δ-values to achieve the desired marginal section at any point in their transmission. The major restriction they experience, naturally, is that M_b's extension and M_f's retraction can only amount to 1 packet per published packet, although they can each be modified to an arbitrary degree in the opposite direction.

4 Acknowledgements

We'd like to thank Dhahsukh Patel for his early support and guidence, Elliott M. Levine for assistance with legibility, and Jai Evans for his support and helpful suggestions.

References

[1] J. Oikarinen and D. Reed. *Internet Relay Chat Protocol.* Network Working Group. 1993. URL: `https://www.rfc-editor.org/rfc/rfc1459`.