How to Use Linear Homomorphic Signature in Network Coding

Li Chen

lichen.xd at gmail.com

Xidian University

September 28, 2013

Outline

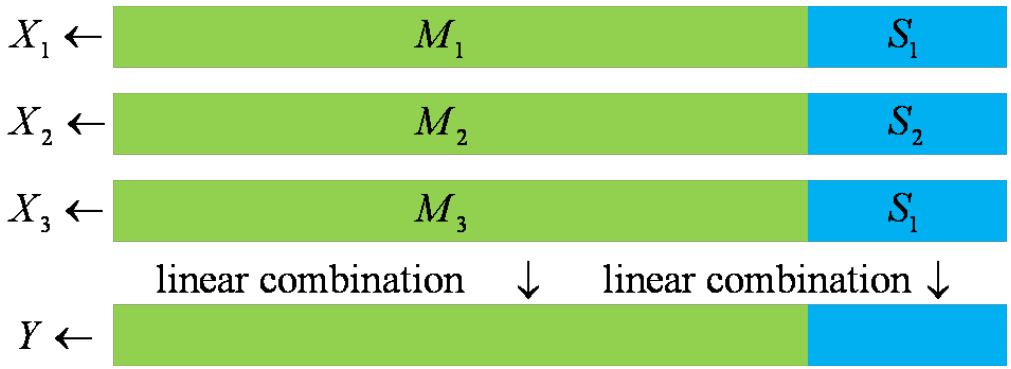
1 Linear Homomorphic Signature

Definition 1.1 A linearly homomorphic signature scheme is a trituple $\mathcal{S} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Veri})$.

- KeyGen: sets $(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathsf{KeyGen}(1^n)$.
- Sign: sets $s \leftarrow \mathsf{Sign}(\mathrm{sk}, m)$. where m is a message, s is the signature.
- Veri: sets $1 \leftarrow \mathsf{Veri}(\mathrm{pk}, m, s)$ if and only if s is a valid signature of m.

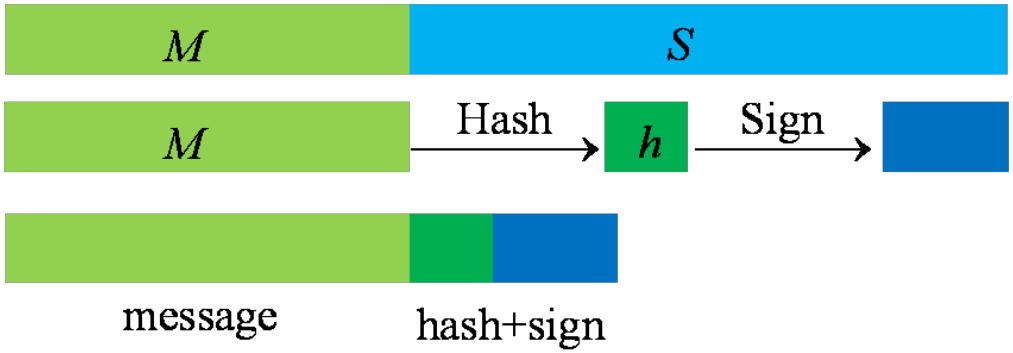
Let F be a field such that $m \in F^n$ and $s \in F^k$, if for arbitrary $m_1, m_2 \in F^n$, $s_1 \leftarrow \mathsf{Sign}(\mathrm{sk}, m_1)$, $s_2 \leftarrow \mathsf{Sign}(\mathrm{sk}, m_2)$, and $c_1, c_2 \in F$, it implies $\mathsf{Sign}(\mathrm{sk}, c_1 m_1 + c_2 m_2) = c_1 s_1 + c_2 s_2$, then we say $\mathcal{S}$ is linearly homomorphic.

Linearly homomorphic signature scheme has a superiority over other signature scheme in network coding. In random network coding, an router just compute a random linear combination of the messages it received and sent this linear combination to a next router. Thus if we use linear signature, a router just compute the same linear combination on the corresponding signatures, and need no extra computation.



One typical linearly homomorphic signature scheme was presented by Boneh[1], it is based on a lattice problem of SIS and enjoys the worst-case security of lattice-based cryptosystems. Bonch's linearly homomorphic signature scheme use $GF(2)$ as the base field, which is very convenient for applying in network coding. But before we use linearly homomorphic signature scheme into network coding, we must solve a key problem.

A randomized signature scheme outputs a signature that has a bigger size than its corresponding message, thus considering the communication overhead we can not sign directly on the ordinary message, but to hash it and sign. Here the main effect of the hash function is compressing the message, then we can sign on the hash value of the ordinary message. The key problem is the hash function and signature scheme must be homomorphic serially, which means the hash function is homomorphic for the operation message, and signature scheme is homomorphic for the operation on hash values.



2 Linearly Homomorphic Hash and Sign

A simple choice is using linearly homomorphic hash function and linearly homomorphic signature scheme, but unfortunatily we can show that there is no collision resistant linearly homomorphic hash function.

Definition 2.1 (Linearly Homomorphic hash) Let H be a hash function from F^n to F^k, where $k < n$. If for arbitrary m_1, m_2 and $h_1 = \mathsf{H}(m_1), h_2 = \mathsf{H}(m_2)$, the following equation holds

$$\mathsf{H}(c_1 m_1 + c_2 m_2) = c_1 h_1 + c_2 h_2$$

where $c_1, c_2 \in F$. Then, we call H a linearly homomorphic hash function.

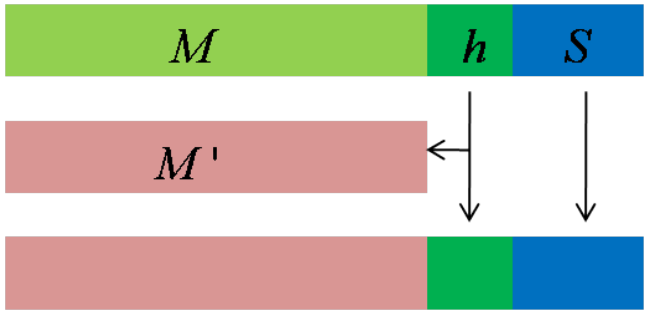
Theorem 2.2 Let H be a linearly homomorphic hash function from F^n to F^k with $k < n$, for an arbitrary $m \in F^n$ and $h = \mathsf{H}(m)$, we can compute a $m' \in F^n$ in polynomial time, such that $h = \mathsf{H}(m')$.

Proof

Since H is linearly homomorphic, we just need to show we can find a $m' \in F^n/0$ in polynomial time, such that $\mathsf{H}(m') = 0$. Let $e_i \in F^n$ with the ith coordinate is 1, and the others are 0. we can compute $h_i = \mathsf{H}(e_i), i \in [1, n]$. Since $h_i \in F^k$, $k < n$, so $h_1, \cdots, h_n$ must be linearly dependent, i.e., there exists $c_1, \cdots, c_n$, not all of which are zero, such that $\sum_{i=1}^{n} c_i h_i = 0$, therefore, $\mathsf{H}(\sum_{i=1}^{n} c_i e_i) = 0$, and $\sum_{i=1}^{n} c_i e_i$ is obviously not zero. □

The following will show that linearly homomorphic hash and sign can not against to pollution attack. To show that, we just need to illuminate a attacker can forge a 'valid' polluted packet, i.e., he can generate a polluted message and a valid hash and signature of this polluted message.

Suppose the attacker obtain a clean message m and its corresponding valid hash value h and signature s, by the assumption that the hash function is linearly homomorphic, then the attacker can compute a message m' such that h and s are the valid hash value and signature of m', here we omit the details that m' can be independent with all clean messages, thus the attacker can use the m' to pollute the whole network.



3 How to Use Linearly Homomorphic Hash and Sign

3.1 Private Hash model

By the analysis of above, if the attacker can not obtain the hash value of any message, then he will not forge a valid signature for a polluted message. So a simple fix method is to share the hash function as a secrete parameter with all legal routers.

3.2 Secrete Channel model

If there exist a secrete channel, we can transmit a secrete key to keep the hash oblivious to any unlegal note. This may be view as a improvement of the secrete channel model of Jaggi et al[2]

3.3 Other Solution

under discussing...

References

References

[1] Dan Boneh and David Mandell Freeman. “Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures.” In: *Public Key Cryptography–PKC 2011*. Springer, 2011, pp. 1–16.

[2] Sidharth Jaggi et al. “Resilient network coding in the presence of byzantine adversaries.” In: *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*. IEEE. 2007, pp. 616–624.

Thanks! & Questions?