On Homomorphic Signatures for Network Coding

Aaram Yun, Jung Hee Cheon, and Yongdae Kim, *Member, IEEE*

Abstract—In this paper, we examine homomorphic signatures that can be used to protect the integrity of network coding. In particular, Yu et al. proposed an RSA-based homomorphic signature scheme recently for this purpose. We show that their scheme in fact does not satisfy the required homomorphic property, and further, even though it can be fixed easily, still it allows no-message forgery attacks.

Index Terms—network coding, homomorphic signature, homomorphic hashing.

1 Introduction

Network coding refers to a new class of routing techniques where a router not only forwards incoming packets but also processes them to produce 'new' outgoing packets. It was originally proposed in Ahlswede et al. [1] as a method for maximizing the information flow of a multicast network, and subsequently found many applications, for example, in sensor networks, wireless networks, and peer-to-peer networks. Li et al. [2] showed that *linear* network coding is sufficient to achieve the maximum throughput of multicast network, and Ho et al. [3] introduced the idea of *random* linear network coding, where each node chooses its own coding coefficients randomly and independently. It has been shown that a random linear network coding achieves the maximum throughput of multicast network with high probability. Also, because of the random and de-central selection of coding, the random linear network coding can be used even when the network topology changes, and can potentially give resilience to random network failures.

For practical application of network coding, protecting integrity of packets is especially important: a malicious node can 'pollute' the entire network by injecting a few false packets. Because of the coding capability of intermediate nodes, any unfiltered false packet will soon propagate to all other nodes, and prevent proper decoding, even when all the other received packets are correct. Note that a conventional cryptographic signature cannot solve this problem, since nodes other than the source not only relay given packets but also 'create' new packets by linear combination. For this reason, some solutions based on the idea of homomorphic hashing or homomorphic signature are proposed [4], [5], [6], [7]. Because these schemes are homomorphic, an intermediate node can combine signatures of incoming packets to form a signature for an outgoing packet, without having the private key of the source.

In INFOCOM 2008, Yu et al. [5] proposed a new homomorphic signature scheme which is based on both discrete logarithm and RSA. In this paper, first we show that in fact their scheme does not satisfy the homomorphic property. The reason for this failure is due to use of two different moduli, and can be fixed by modifying the scheme to use a single modulus. We show that this modified scheme is also not secure, by showing that an attacker can successfully mount a no-message attack, and more powerful single-message attack.

2 Network Coding Security

First, let us describe linear network coding briefly. Let $G = (V, E)$ be a directed graph. Suppose a source $s \in V$ wants to send a large file F to a set $T \subseteq V$ of clients. We assume that the file F is represented as a sequence of m vectors $\bar{\mathbf{f}}_1, \ldots, \bar{\mathbf{f}}_m \in \mathbb{F}^n$, where $\mathbb{F}^n$ is the n dimensional vector space over a finite field $\mathbb{F}$. Then the source creates *augmented vectors* of $\bar{\mathbf{f}}_i = (\bar{f}_{i,1}, \ldots, \bar{f}_{i,n})$ by setting

$$\mathbf{f}_i \stackrel{\text{def}}{=} (\bar{f}_{i,1}, \ldots, \bar{f}_{i,n}, \overbrace{\underbrace{0, \ldots, 0, 1}_{i}, 0, \ldots, 0}^{m}),$$

that is, each augmented vector $\mathbf{f}_i$ is of dimension $t \stackrel{\text{def}}{=} m+n$, and the last m entries are all zero except at $(n+i)$-th, where it is 1.

The source then propagates linear combinations (over $\mathbb{F}$) of the augmented vectors to other nodes. Each node, after receiving packets $\mathbf{v}_1, \ldots, \mathbf{v}_l \in \mathbb{F}^t$ from its l incoming channels, computes a linear combination

$$\mathbf{w}_i = \sum_{j=1}^{l} \alpha_{i,j} \mathbf{v}_j,$$

for some $\alpha_{i,j} \in \mathbb{F}$, and transmits $\mathbf{w}_i$ to its i-th outgoing channel. Therefore, assuming that all transmissions are

- *Aaram Yun and Yongdae Kim are with the Department of Computer Science & Engineering, University of Minnesota, Minneapolis, MN 55455. E-mail: {aaram,kyd}@cs.umn.edu*
- *Jung Hee Cheon is with the Department of Mathematics, Seoul National University, 599 Gwanangno, Gwanak-gu, Seoul 151-747, Korea. E-mail: jhcheon@snu.ac.kr*

done without error, each packet is a linear combination of the original augmented vectors $\mathbf{f}_1, \ldots, \mathbf{f}_m$ of the file over $\mathbb{F}$.

The coefficients $\alpha_{i,j}$ can be determined centrally given a static network topology, or can be randomly chosen by each nodes in the case of *random* linear network coding. It has proven that a randomly chosen linear network coding can achieve the optimal network performance with high probability. Moreover, because coefficients are chosen randomly, it is also usable even when the network topology changes.

3 Correctness of the scheme of Yu et al.

In [5], Yu et al. proposed a homomorphic signature scheme for network coding. We will quickly describe their scheme: two primes p, q satisfying $q \mid p-1$ are chosen. Let G be the subgroup of order q in $\mathbb{Z}_p^*$. Then, $t = m + n$ elements $g_1, \ldots, g_t$ are chosen from G. Let N be an RSA modulus of same bit length as p, i.e., N is a product of two primes of the same bit length. Choose e and d such that $ed \equiv 1 \pmod{\phi(N)}$. Then the public key is $PK = (p, q, g_1, \ldots, g_t, N, e)$, and the private key is $SK = d$. The base field for the network coding operation is $\mathbb{F} = \mathbb{Z}_q$.

Given a packet $\mathbf{v} = (v_1, \ldots, v_t)$, the signature is calculated as

$$\mathsf{Sig}(SK, \mathbf{v}) \stackrel{\text{def}}{=} \left(\prod_{j=1}^{t} g_j^{v_j} \bmod p \right)^d \bmod N.$$

And given a packet $\mathbf{v}$ and a corresponding signature σ, the verification $\mathsf{Vf}(PK, \mathbf{v}, \sigma)$ can be done by checking

$$\sigma^e \stackrel{?}{\equiv} \left(\prod_{j=1}^{t} g_j^{v_j} \bmod p \right) \pmod N.$$

Note that $\mathsf{Vf}(PK, \mathbf{v}, \mathsf{Sig}(SK, \mathbf{v})) = \mathsf{true}$ holds.

They claim that this is a homomorphic signature scheme. Specifically, if σ_i is a valid signature for a packet $\mathbf{v}_i$, and if $\mathbf{w} = \sum_i \alpha_i \mathbf{v}_i$ for some $\alpha_i \in \mathbb{Z}_q$, then $\tau = \prod_i \sigma_i^{\alpha_i} \bmod N$ is supposed to be a valid signature for $\mathbf{w}$.

However, it is easy to see that this does not hold: consider the simplest case of $\mathbf{w} = 2\mathbf{v}$. Let σ be the signature for $\mathbf{v}$. Then $\sigma^2 \bmod N$ should be a valid signature for $\mathbf{w}$. This means that, if $\sigma^e \equiv \left(\prod_{j=1}^{t} g_j^{v_j} \bmod p \right) \pmod N$, then $\sigma^{2e} \equiv \left(\prod_{j=1}^{t} g_j^{2v_j} \bmod p \right) \pmod N$. Letting $X = \sigma^e$ and $Y = \prod_{j=1}^{t} g_j^{v_j} \bmod p$, this means that if $X \equiv Y \pmod N$, then $X^2 \equiv (Y^2 \bmod p) \pmod N$, which cannot be true in general.

This can be also seen from a small example: consider the case of $t = 4$, $N = 143 = 11 \cdot 13$, $p = 139 = 1 + 2 \cdot 3 \cdot 23$, $q = 23$, $e = 7$, $d = 103$, and $(g_1, g_2, g_3, g_4) = (55, 64, 65, 129)$. This set of parameters satisfy all the requirements of the parameter setup for the scheme. If $\mathbf{v} = (1, 0, 1, 0)$, then the signature σ is

$$\begin{aligned}(g_1 g_3 \bmod p)^d \bmod N &= (55 \cdot 65 \bmod 139)^{103} \bmod 143 \\ &= 100^{103} \bmod 143 \\ &= 100\end{aligned}$$

Then $\sigma' = \sigma^2 \bmod N$, which is $100^2 \bmod 143 = 133$ should be a valid signature of $\mathbf{w} = 2\mathbf{v} = (2, 0, 2, 0)$. This means that

$$133^7 \stackrel{?}{\equiv} (55 \cdot 65)^2 \bmod 139 \pmod{143}.$$

But $133^7 \bmod 143 = 133$, and on the other hand,

$$(55 \cdot 65)^2 \bmod 139 \bmod 143 = 131 \bmod 143 = 131.$$

4 A simple fix, and more forgeries

The signature scheme of Yu et al. is constructed analoguous to the traditional 'hash-and-sign' paradigm: it is a composition of a homomorphic hash function $\mathbf{v} \mapsto \prod g_j^{v_j} \bmod p$ and the 'bare' RSA signature scheme $x \mapsto x^d \bmod N$. In fact, this is very similar to the Full Domain Hash [8], which is $H(M)^d \bmod N$, except that instead of a regular hash function like SHA-1, a homomorphic hash function is used for retaining the homomorphic property.

Essentially, the reason for the failure of homomorphic property of the scheme of Yu et al. is because they use two different moduli p and N: both the inner hash function and the outer RSA signature are homomorphic, but they are homomorphic with respect to different, incompatible operations. Therefore, one simple way to regain the homomorphic property is to let '$p = N$':

$$\mathsf{Sig}(SK, \mathbf{v}) \stackrel{\text{def}}{=} \left(\prod_{j=1}^{t} g_j^{v_j} \right)^d \bmod N.$$

This modified scheme is clearly homomorphic. Unfortunately, we can show that this scheme is insecure by exhibiting a simple no-message attack, that is, the attacker can make a successful forgery based only on the public key and public parameters. Consider the packet $\mathbf{v} = (e, 0, 0, \ldots, 0)$. Then $\mathsf{Sig}(SK, \mathbf{v}) = (g_1^e)^d \bmod N = g_1$. Therefore g_1 is a valid signature for $\mathbf{v} = (e, 0, 0, \ldots, 0)$, but on the other hand this $\mathbf{v}$ clearly does not belong to the subspace spanned by the augmented vectors of the original file.

One way to cope with this problem might be to restrict the message space to vectors of entries smaller than the public exponent e. This would eliminate one benefit of RSA signatures, namely, we can pick e to be very small for faster verification. Here, we need to pick e large enough so that all packets are smaller than e given the size of the network.

Still, even this version can be easily broken: the attacker has to eavesdrop at least one packet $\mathbf{v} = (v_1, \ldots, v_t)$ and its signature σ. The attacker then forms yet another packet $\mathbf{w} = (w_1, \ldots, w_t) \stackrel{\text{def}}{=} \alpha\mathbf{v} +$

$(\beta_1 e, \ldots, \beta_t e)$. Clearly, $\tau \stackrel{\text{def}}{=} \sigma^\alpha \cdot g_1^{\beta_1} \cdots g_t^{\beta_t}$ is a valid signature of $\mathbf{w}$. The problem is to select α and β_i's so that all of $w_i = \alpha v_i + \beta_t e$ are greater than or equal to 0 and less than e, which is equivalent to

$$0 \leq \alpha \frac{v_i}{e} + \beta_i < 1.$$

We see that for each choice of α, there is one unique β_i satisfying the above. Pick α, which should not be divisible by e, large enough so that at least some of β_i are nonzero. This produces a successful forgery.

5 Concluding Remarks

In this paper, we pointed out that the signature scheme of Yu et al. is in fact not homomorphic. This is due to the use of two different moduli, which gives a composition of an homomorphic hash function and a homomorphic signature, with incompatible group structures. We further examined the security of the modified scheme where only one modulus is used. We found out that this was not enough to produce a secure homomorphic signature scheme. The main reason for this failure was that the scheme allows a trivial no-message attack. Combined with the homomorphic property, this leads to even more stronger forgery when the attacker eavesdrops a few packets.

This is unfortunate, because the idea of Yu et al. is a natural direction to design an RSA-based homomorphic signature; if we pursue the approach of following 'hash-and-sign' paradigm but using a homomorphic hash function instead of a regular cryptographic hash function, there are not many choices for hash functions other than ones involving group operations, like in the scheme of Yu et al. For the signature scheme, one possibility is to use the BLS short signature [9], and indeed this combination is used to construct the homomorphic signature scheme of Boneh et al. [4]. If the combination of the homomorphic hash function and the plain RSA signature would have been secure, it would have all the traditional benefits of RSA schemes, like faster signature verification and simpler implementation. It would be an interesting open problem to design a secure RSA-based homomorphic signature scheme.

Acknowledgements

The first and the third authors were supported, in part, by the US National Science Foundation (NFS) grants CCF-0621462 and CNS-0716025. The second author was supported by NAP of Korea Research Council of Fundamental Science & Technology.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[2] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[3] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proceedings of IEEE International Symposium on Information Theory*, 2003, p. 442.

[4] D. Boneh, D. Freeman, J. Katz, and B. Waters, "Signing a linear subspace: Signature schemes for network coding," in *Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography (PKC)*, 2009, to be published: available at http://eprint.iacr.org/2008/316.

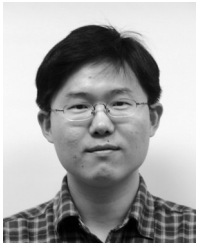
[5] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient signature-based scheme for securing network coding against pollution attacks," in *Proceedings of the 27th Conference on Computer Communications (INFOCOM)*. IEEE, April 2008, pp. 1409–1417.

[6] D. Charles, K. Jain, and K. Lauter, "Signatures for network coding," in *Proceedings of the 40th Annual Conference on Information Sciences and Systems (CISS)*, March 2006, pp. 857–863.

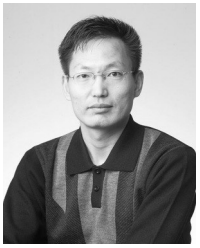
[7] M. N. Krohn, M. J. Freedman, and D. Mazières, "On-the-fly verification of rateless erasure codes for efficient content distribution," in *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2004, pp. 226–240.

[8] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, 1993, pp. 62–73.

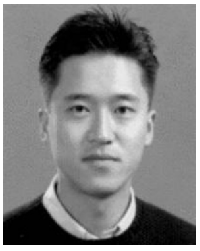
[9] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *Journal of Cryptology*, vol. 17, no. 4, pp. 297–319, September 2004.



Aaram Yun Aaram Yun received his BS degree in mathematics from Korea Advanced Institute of Science and Technology (KAIST), and the Ph.D. degree in mathematics from Yale University. He is currently working as a postdoctoral researcher in University of Minnesota. Before then, he was at Electronics and Telecommunication Research Institute (ETRI) as a senior researcher. His research interests include applied cryptography and information security.



Jung Hee Cheon Jung Hee Cheon received his BS, MS, and Ph.D. degrees in mathematics from Korea Advanced Institute of Science and Technology (KAIST). He is an associate professor in the Department of Mathematical Sciences at Seoul National University (SNU). Prior to SNU, he was in Electronics and Telecommunication Research Institute (ETRI) as a senior researcher, Brown University as a visiting scientist, and Information and Communications University (ICU) as an assistant professor. His research interests include computational number theory, cryptography and information security.



Yongdae Kim Yongdae Kim is an associate professor at the University of Minnesota - Twin Cities. He received PhD degree from University of Southern California and joined University of Minnesota in 2002. He has been working on various projects in data and communication security. He received NSF career award on storage security and McKnight Land-Grant Professorship Award from University of Minnesota in 2005. His research interests include security issues for distributed systems such as P2P systems, storage systems, sensor and ad hoc networks.