

Signatures for Network Coding

CISS 2006, Princeton.

Denis Charles

Kamal Jain

Kristin Lauter

Microsoft Research, Redmond.

Network coding

Let $G = (V, E)$ be a directed graph. A source $s \in V$ wants to transmit a file D to $T \subseteq V$. Let the out-degree of s be k , and assume that the in-degree of every $t \in T$ is also k .

Treat D as w_1, \dots, w_k vectors in \mathbb{F}_q^d .

The source constructs (and transmits) the augmented vectors:

$$\begin{aligned} \mathbf{v}_1 &= \langle \underbrace{1, 0, \dots, 0}_k, w_{11}, \dots, w_{1d} \rangle \\ \mathbf{v}_2 &= \langle 0, 1, \dots, 0, w_{21}, \dots, w_{2d} \rangle \\ &\vdots \\ \mathbf{v}_k &= \langle 0, 0, \dots, 1, w_{k1}, \dots, w_{kd} \rangle. \end{aligned}$$

For an edge e let $\mathbf{y}(e)$ denote the vector transmitted along that edge.

At each edge $e \in E$ the vectors are combined as follows:

$$\mathbf{y}(e) = \sum_{f \in E: \text{out}(f) = \text{in}(e)} m_e(f) \mathbf{y}(f).$$

where $m_e(f) \in \mathbb{F}_q$ are picked at random.

Decoding at the receiver

Each receiver, $t \in T$, gets k vectors y_1, \dots, y_k which are **random** linear combinations of the v_i 's.

In fact, if

$$y_i = \langle \alpha_{i1}, \dots, \alpha_{ik}, a_{i1}, \dots, a_{id} \rangle$$

then

$$y_i = \sum_{1 \leq j \leq k} \alpha_{ij} v_j.$$

Thus we can invert the linear transformation to find the v_i 's with high probability.

★ Network coding has been shown to optimally use bandwidth in a network, maximizing information flow.

but...

★ The scheme is very vulnerable to pollution attacks. A node injecting garbage can quickly affect many receivers.

Possible fixes

- ★ Digital signature on the whole file... does not detect pollution quickly.
- ★ Sign each vector... need to contact the source to sign linear combinations.

Homomorphic Hashing

[Krohn, Freedman, Mazières '04] Suppose we have a hash function $H: V \rightarrow G$ such that:

- ★ H is **collision resistant** – it is hard to find x and y such that $H(x) = H(y)$;
- ★ H is a **homomorphism** – $H(x + y) = H(x)H(y)$.

Then server can securely distribute $H(\mathbf{v}_i)$ to each receiver, and to check if

$$\mathbf{y} = \sum_{1 \leq i \leq k} \alpha_i \mathbf{v}_i$$

we can check if

$$H(\mathbf{y}) \stackrel{?}{=} \sum_{1 \leq i \leq k} \alpha_i H(\mathbf{v}_i).$$

Problems: Server needs to transfer secure information to each of the receivers. H is expensive to compute.

Secure random checksums

[Gkantsidis, Rodriguez '06] Each user, u , securely gets from the server:

★ A linear functional $\lambda_u : V \rightarrow \mathbb{F}_q$ – a vector \mathbf{w} such that

$$\lambda_u(\mathbf{v}) = \mathbf{w} \cdot \mathbf{v};$$

★ Also gets $\lambda_u(\mathbf{v}_i)$ for $1 \leq i \leq k$.

Verification: The claim that

$$\mathbf{y} = \sum_{1 \leq i \leq k} \alpha_i \mathbf{v}_i$$

can be checked by

$$\lambda_u(\mathbf{y}) \stackrel{?}{=} \sum_{1 \leq i \leq k} \alpha_i \lambda_u(\mathbf{v}_i).$$

Problems: Server needs to compute $\lambda_u(\mathbf{v}_i)$ for **each** user. Also, secure communication is costly. A random vector will pass the test with probability $1/|\mathbb{F}_q|$.

Advantage: Verification is very fast.

Our Scheme

- ★ Operates in a public key cryptography infrastructure.
- ★ It is a homomorphic signature scheme based on elliptic curves.

Elliptic curves over a finite field

Let \mathbb{F}_q be a finite field such that q is not a power of 2 or 3. Then an elliptic curve E over \mathbb{F}_q is a curve given by an equation of the form

$$y^2 = x^3 + ax + b,$$

where $a, b \in \mathbb{F}_q$ such that $4a^3 + 27b^2 \neq 0$. Let $K \supseteq \mathbb{F}_q$, then

$$E(K) = \{(x, y) \mid y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}$$

forms an abelian group with \mathcal{O} as identity. The group operations can be performed efficiently.

Notation: If m is any integer

$$E[m] = \{P \in E(\overline{\mathbb{F}_q}) : mP = \mathcal{O}\}.$$

Weil pairing

Fact: If E/\mathbb{F}_q is an elliptic curve and $\gcd(m, q) = 1$ then

$$E[m] \cong (\mathbb{Z}/m\mathbb{Z}) \times (\mathbb{Z}/m\mathbb{Z})$$

There is a map $e_m : E[m] \times E[m] \rightarrow \mu_m(\mathbb{F}_q)$ such that

★ (Bilinear) $e_m(P + R, Q) = e_m(P, Q)e_m(R, Q)$,
 $e_m(P, Q + R) = e_m(P, Q)e_m(P, R)$;

★ (Non-degenerate) $e_m(P, Q) = 1$ for all P implies that $Q = \mathcal{O}$;

★ (Alternating) $e_m(P, P) = 1$.

Also, e_m can be computed efficiently!

Homomorphic signatures

Let p be a prime and q a prime power. Let V/\mathbb{F}_p be a vector space of dimension D and E/\mathbb{F}_q be an elliptic curve such that

$$P_1, \dots, P_D \in E[p].$$

Define $h : V \rightarrow E[p]$ as follows:

$$h(\langle u_1, \dots, u_D \rangle) = \sum_{1 \leq i \leq D} u_i P_i.$$

The function h is a homomorphism from V to $E[p]$.

The server chooses s_1, \dots, s_D secretly in \mathbb{F}_p and publishes

$$P_i, s_i Q \text{ for } 1 \leq i \leq D$$

and also another point Q of p -torsion such that $e_p(P_i, Q) \neq 1$.

The **signature** of the vector $\mathbf{v} = \langle u_1, \dots, u_D \rangle$ is

$$\sigma(\mathbf{v}) = \sum_{1 \leq i \leq D} u_i s_i P_i.$$

Note: This signature is homomorphic!

Signature verification

Given $\mathbf{v} = \langle u_1, \dots, u_D \rangle$ and its signature σ verify that

$$\begin{aligned} e(\sigma, Q) &= e\left(\sum_{1 \leq i \leq D} u_i s_i P_i, Q\right) \\ &= \prod_i e(u_i s_i P_i, Q) \\ &\stackrel{?}{=} \prod_i e(u_i P_i, s_i Q). \end{aligned}$$

System setup

The server computes $\sigma(\mathbf{v}_i)$ for each $1 \leq i \leq k$. Transmits $\mathbf{v}_i, \sigma(\mathbf{v}_i)$.
At each edge e while computing

$$\mathbf{y}(e) = \sum_{f \in E: \text{out}(f)=\text{in}(e)} \mathbf{m}_e(f) \mathbf{y}(f)$$

also compute

$$\sigma(\mathbf{y}(e)) = \sum_{f \in E: \text{out}(f)=\text{in}(e)} \mathbf{m}_e(f) \sigma(\mathbf{y}(f))$$

on the elliptic curve E .

Transmission overhead is $2 \log q$ bits (to transmit x and y coordinates of the $\sigma(\mathbf{v}_i)$). Computation of new signature, $\sigma(\mathbf{y}(e))$, requires $O(d_e)$ exponentiations in \mathbb{F}_q .

The amount of public information is $O(D \log q)$ bits.

Security

Attacker can either

produce a collision under the hash function...

which is as hard as computing discrete logs on the elliptic curve;

or forge the signature...

which is as hard as solving computational co-Diffie-Hellman problem on the elliptic curve.

The computational co-Diffie-Hellman problem

Let G_1, G_2 be two groups, given $g, g^a \in G_2$ and $h \in G_1$ compute $h^a \in G_1$.

In our context G_1 and G_2 are the two pieces of the p -torsion points.

$$E[p] \cong \underbrace{(\mathbb{Z}/p\mathbb{Z})}_{G_1} \times \underbrace{(\mathbb{Z}/p\mathbb{Z})}_{G_2}.$$

Outline of proof of collision resistance

Given P_1, \dots, P_r points in $E[p]$ find

$$\mathbf{a} = \langle a_1, \dots, a_r \rangle \in \mathbb{F}_p^r$$

and

$$\mathbf{b} = \langle b_1, \dots, b_r \rangle \in \mathbb{F}_p^r$$

such that $\mathbf{a} \neq \mathbf{b}$ and

$$\sum_{1 \leq i \leq r} a_i P_i = \sum_{1 \leq j \leq r} b_j P_j.$$

If $r = 2$ then we get $xP + yQ = uP + vQ$. Thus $(x - u)P + (y - v)Q = 0$.

We must have $x \neq u$ and $y \neq v$ thus

$$Q = -(x - u)(y - v)^{-1}P.$$

For $r > 2$, take $P_1 = r_1P$ and $P_i = r_iQ$ for $i \geq 2$ where r_i are picked at random.

A collision yields

$$ar_1P + \left(\sum_{2 \leq i \leq r} b_i r_i \right) Q = 0.$$

As long as $\sum_{2 \leq i \leq r} b_i r_i \neq 0 \pmod{p}$ we can solve for the discrete log of Q .

Since r_i are unknown to the oracle, we can interchange the order of this process: For a given sequence of b_i , what is the probability that the r_i 's we picked satisfy:

$$\sum_{2 \leq i \leq r} b_i r_i = 0?$$

and this is $1/p$. \square

Advantages of the system

1. Establishes authentication in addition to detecting pollution.
2. No need for distributing secure hash digests.
3. Smaller bit lengths suffice. Signatures of length 180 bits have as much security as 1024 bit RSA signatures.
4. Public information does not change for subsequent file transmission.

Some timing information

Implementation in C++ on an Opteron 252 2.6Ghz processor.

Vector operations on V/\mathbb{F}_p where p is 192-bits and $\dim V = 50$ requires 8×10^{-5} seconds.

Weil pairing computation on E/\mathbb{F}_p takes 4 milliseconds.

In general the complexity is $O(\log^{2+\epsilon} q)$ bit operations for verification.

An optimization for signature verification

Given $v = \langle u_1, \dots, u_D \rangle$ and its signature σ verify that

$$\begin{aligned} e(\sigma, Q) &\stackrel{?}{=} \prod_i e(u_i P_i, s_i Q) \\ &= \prod_i e(P_i, s_i Q)^{u_i} \\ &= \prod_i \zeta_i^{u_i} \end{aligned}$$

where $\zeta_i = e(P_i, s_i Q)$ are pre-computed.

So verification of signature requires only **one** Weil pairing computation.

Open problem

Design a (provably secure) homomorphic signature scheme such that signatures are fast to verify!