Honey Badger Byzantine

Videos

https://youtu.be/jMePVYNH_Rg - Lect-34-4010-pt1-Threshold-Signatures.mp4
https://youtu.be/H_MNY4ro_sM - Lect-34-4010-pt2-crypto-currencies.mp4

From Amazon S3 - for download (same as youtube videos)

http://uw-s20-2015.s3.amazonaws.com/Lect-34-4010-pt1-Threshold-Signatures.mp4
http://uw-s20-2015.s3.amazonaws.com/Lect-34-4010-pt2-crypto-currencies.mp4

Signature

Signature: Step 1 and 2

We have α, which is a t-ECDSA private key in encrypted form shared between all the signers, and we have y which is a t-ECDSA public key. Public keys are just a point on the elliptic curve.

In the first round each party draws a random integer ρ:

t-ECDSA Signature

Round 1 and 2

Using the commit and reveal pattern, each player P_i:

- selects a random ρ_i in Z_q
- computes $u_i = E(\rho_i)$
- computes $v_i = \rho_i \times_e \alpha = E(\rho_i x)$
- provides ZKP which states u_i and v_i are correct

After the round 2, all players join shares together:

- $u = \sum u_i = E(\rho)$
- $v = \sum v_i = E(\rho x)$

Encrypt the value with additively homomorphic threshold encryption scheme:

t-ECDSA Signature

Round 1 and 2

Using the commit and reveal pattern, each player P_i:

- selects a random ρ_i in Z_q
- computes $u_i = E(\rho_i)$
- computes $v_i = \rho_i \times_e \alpha = E(\rho_i x)$
- provides ZKP which states u_i and v_i are correct

After the round 2, all players join shares together:

- $u = \sum u_i = E(\rho)$
- $v = \sum v_i = E(\rho x)$

Multiply the secret ECDSA key (α below) by this random value:

t-ECDSA Signature

Round 1 and 2

Using the commit and reveal pattern, each player P_i:

- selects a random ρ_i in Z_q
- computes $u_i = E(\rho_i)$
- computes $v_i = \rho_i \times_e \alpha = E(\rho_i x)$
- provides ZKP which states u_i and v_i are correct

After the round 2, all players join shares together:

- $u = \sum u_i = E(\rho)$
- $v = \sum v_i = E(\rho x)$

Use addition to implement multiplication. Each signer publishes commitment to those values, and in the second round reveals all those values, along with the zero-knowledge proof, stating that they make sense:

t-ECDSA Signature

Round 1 and 2

Using the commit and reveal pattern, each player P_i:

- selects a random ρ_i in Z_q
- computes $u_i = E(\rho_i)$
- computes $v_i = \rho_i \times_e \alpha = E(\rho_i x)$
- provides ZKP which states u_i and v_i are correct

After the round 2, all players join shares together:

- $u = \sum u_i = E(\rho)$
- $v = \sum v_i = E(\rho x)$

Using zero-knowledge proof and the commitment we can now prove that the values we have are correct.

Join all the shares together (This means broadcast all the shares so that everybody has them):

Reveal the commitments:

t-ECDSA Signature

Round 3 and 4

Using the commit and reveal pattern, each player P_i:

- selects a random k_i in Z_q
- selects a random c_i from $[-q^6, q^6]$
- computes $r_i = g^{k_i}$
- computes $w_i = E(k_i \rho + c_i q)$
- provides ZKP which states r_i and w_i are correct

After the round 4, all players join shares together:

- $k = \sum k_i$
- $c = \sum c_i$
- $w = \sum w_i = E(k\rho + cq)$
- $r = H(\sum r_i) = H(g^k)$

After the round 2, all players know those values:

- $u = \sum u_i = E(\rho)$ $\quad u_i \in Z_q$
- $v = \sum v_i = E(\rho x)$

Signature: Step 3 and 4

The same commit-reveal pattern is used in the 3rd and 4th round:

t-ECDSA Signature

Round 3 and 4

Using the commit and reveal pattern, each player P_i:

- selects a random k_i in Z_q
- selects a random c_i from $[-q^6, q^6]$
- computes $r_i = g^{k_i}$
- computes $w_i = E(k_i \rho + c_i q)$
- provides ZKP which states r_i and w_i are correct

After the round 4, all players join shares together:

- $k = \sum k_i$
- $c = \sum c_i$
- $w = \sum w_i = E(k\rho + cq)$
- $r = H(\sum r_i) = H(g^k)$

After the round 2, all players know those values:

- $u = \sum u_i = E(\rho)$ $\quad u_i \in Z_q$
- $v = \sum v_i = E(\rho x)$

On the right side we have all the parameters that were evaluated so far, and all players have the same values.

In the third round each party draws a random integer `k_i` :

t-ECDSA Signature

Round 3 and 4

Using the commit and reveal pattern, each player P_i:

- selects a random k_i in Z_q
- selects a random c_i from $[-q^6, q^6]$
- computes $r_i = g^{k_i}$
- computes $w_i = E(k_i \rho + c_i q)$
- provides ZKP which states r_i and w_i are correct

After the round 4, all players join shares together:

- $k = \sum k_i$
- $c = \sum c_i$
- $w = \sum w_i = E(k\rho + cq)$
- $r = H(\sum r_i) = H(g^k)$

After the round 2, all players know those values:

- $u = \sum u_i = E(\rho)$ $\quad u_i \in Z_q$
- $v = \sum v_i = E(\rho x)$

And a random integer `c_i` :

t-ECDSA Signature

Round 3 and 4

Using the commit and reveal pattern, each player P_i:

- selects a random k_i in Z_q
- selects a random c_i from $[-q^6, q^6]$
- computes $r_i = g^{k_i}$
- computes $w_i = E(k_i \rho + c_i q)$
- provides ZKP which states r_i and w_i are correct

After the round 4, all players join shares together:

- $k = \sum k_i$
- $c = \sum c_i$
- $w = \sum w_i = E(k\rho + cq)$
- $r = H(\sum r_i) = H(g^k)$

After the round 2, all players know those values:

- $u = \sum u_i = E(\rho)$ $\quad u_i \in Z_q$
- $v = \sum v_i = E(\rho x)$

`q` all the time stands for the cardinality of the elliptic curve, so the number of points on the elliptic curve.

Each party computes `r_i` as `g` to the power of `k` —we basically multiply curve generator point by it:

t-ECDSA Signature

Round 3 and 4

Using the commit and reveal pattern, each player P_i:
- selects a random k_i in Z_q
- selects a random c_i from $[-q^6, q^6]$
- computes $r_i = g^{k_i}$
- computes $w_i = E(k_i \rho + c_i q)$
- provides ZKP which states r_i and w_i are correct

After the round 2, all players know those values:
- $u = \sum u_i = E(\rho)$ $\quad u_i \in Z_q$
- $v = \sum v_i = E(\rho x)$

After the round 4, all players join shares together:
- $k = \sum k_i$
- $c = \sum c_i$
- $w = \sum w_i = E(k\rho + cq)$
- $r = H(\sum r_i) = H(g^k)$

Each party computes the parameter `w` which is `k` time `ρ` plus `c` times `q`:

t-ECDSA Signature

Round 3 and 4

Using the commit and reveal pattern, each player P_i:
- selects a random k_i in Z_q
- selects a random c_i from $[-q^6, q^6]$
- computes $r_i = g^{k_i}$
- computes $w_i = E(k_i \rho + c_i q)$
- provides ZKP which states r_i and w_i are correct

After the round 2, all players know those values:
- $u = \sum u_i = E(\rho)$ $\quad u_i \in Z_q$
- $v = \sum v_i = E(\rho x)$

After the round 4, all players join shares together:
- $k = \sum k_i$
- $c = \sum c_i$
- $w = \sum w_i = E(k\rho + cq)$
- $r = H(\sum r_i) = H(g^k)$

`q` all the time is the cardinality of the elliptic curve, and we can compute it because we use additively homomorphic threshold encryption.

At the end, each party commits to all those parameters, and in the round 4 generated parameters are revealed, along with the zero-knowledge proof stating that they make sense together:

t-ECDSA Signature

Round 3 and 4

Using the commit and reveal pattern, each player P_i:
- selects a random k_i in Z_q
- selects a random c_i from $[-q^6, q^6]$
- computes $r_i = g^{k_i}$
- computes $w_i = E(k_i \rho + c_i q)$
- provides ZKP which states r_i and w_i are correct

After the round 2, all players know those values:
- $u = \sum u_i = E(\rho)$ $\quad u_i \in Z_q$
- $v = \sum v_i = E(\rho x)$

After the round 4, all players join shares together:
- $k = \sum k_i$
- $c = \sum c_i$
- $w = \sum w_i = E(k\rho + cq)$
- $r = H(\sum r_i) = H(g^k)$

Having all those parameters from all the group members we can add them together, just like we did after the round 2. We sum up all `k` shares, all `c` shares, all `w` shares. We evaluate parameter `r` as a sum of all `r_i` shares and we use a special hash function:

t-ECDSA Signature

Round 3 and 4

Using the commit and reveal pattern, each player P_i:
- selects a random k_i in Z_q
- selects a random c_i from $[-q^6, q^6]$
- computes $r_i = g^{k_i}$
- computes $w_i = E(k_i \rho + c_i q)$
- provides ZKP which states r_i and w_i are correct

After the round 2, all players know those values:
- $u = \sum u_i = E(\rho)$ $\quad u_i \in Z_q$
- $v = \sum v_i = E(\rho x)$

After the round 4, all players join shares together:
- $k = \sum k_i$
- $c = \sum c_i$
- $w = \sum w_i = E(k\rho + cq)$
- $r = H(\sum r_i) = H(g^k)$

This is the standard ECDSA. `x` is a point coordinate modulo the `q` elliptic curve order.

Signature: Step 5

All parameters on the right side are shared by all signers in the group.

Now we need to do some discrete mathematics magic to produce a signature. Using all those parameters we have evaluated so far, and since we operate on encrypted data, the signature will be also encrypted. But this is something we will deal in the final round 6.

t-ECDSA Signature

Round 5

All players jointly decrypt w:
$\eta = TDec(w) = k\rho \bmod q$

and compute:
$\Psi = \eta^{-1} \bmod q = k^{-1}\,\rho^{-1}$

$$\begin{aligned}\sigma &= \Psi \times_e [(m \times_e u) +_e (r \times_e v)] \\ &= \Psi \times_e [E(m\rho) +_e E(r\rho x)] \\ &= (k^{-1}\,\rho^{-1}) \times_e [E(\rho(m + xr))] \\ &= E(k^{-1}\,(m + xr)) \\ &= E(s)\end{aligned}$$

After round 4, all players know those values:

- $u = \sum u_i = E(\rho)$ $\quad u_i \in Z_q$
- $v = \sum v_i = E(\rho x)$
- $k = \sum k_i$ $\quad k_i \in Z_q$
- $c = \sum c_i$ $\quad c_i \in [-q^6, q^6]$
- $w = \sum w_i = E(k\rho + cq)$
- $r = H(\sum r_i) = H(g^k)$

The very first thing we need to do is that we execute a threshold decryption mechanism to have all the players decrypt the parameter `w` and assign this value to η :

t-ECDSA Signature

Round 5

All players jointly decrypt w:
$\eta = TDec(w) = k\rho \bmod q$

and compute:
$\Psi = \eta^{-1} \bmod q = k^{-1}\,\rho^{-1}$

$$\begin{aligned}\sigma &= \Psi \times_e [(m \times_e u) +_e (r \times_e v)] \\ &= \Psi \times_e [E(m\rho) +_e E(r\rho x)] \\ &= (k^{-1}\,\rho^{-1}) \times_e [E(\rho(m + xr))] \\ &= E(k^{-1}\,(m + xr)) \\ &= E(s)\end{aligned}$$

After round 4, all players know those values:

- $u = \sum u_i = E(\rho)$ $\quad u_i \in Z_q$
- $v = \sum v_i = E(\rho x)$
- $k = \sum k_i$ $\quad k_i \in Z_q$
- $c = \sum c_i$ $\quad c_i \in [-q^6, q^6]$
- $w = \sum w_i = E(k\rho + cq)$
- $r = H(\sum r_i) = H(g^k)$

Compute yet one parameter called Ψ which is multiplicative inverse of η modulo q , and q is all the time cardinality of the elliptic curve:

t-ECDSA Signature

Round 5

All players jointly decrypt w:
$\eta = TDec(w) = k\rho \bmod q$

and compute:
$\Psi = \eta^{-1} \bmod q = k^{-1}\ \rho^{-1}$

$$\begin{aligned}\sigma &= \Psi \times_e [(m \times_e u) +_e (r \times_e v)] \\ &= \Psi \times_e [E(m\rho) +_e E(r\rho x)] \\ &= (k^{-1}\ \rho^{-1}) \times_e [E(\rho(m + xr))] \\ &= E(k^{-1}\ (m + xr)) \\ &= E(s)\end{aligned}$$

After round 4, all players know those values:

- $u = \sum u_i = E(\rho)$ $\quad u_i \in Z_q$
- $v = \sum v_i = E(\rho x)$
- $k = \sum k_i$ $\quad k_i \in Z_q$
- $c = \sum c_i$ $\quad c_i \in [-q^6, q^6]$
- $w = \sum w_i = E(k\rho + cq)$
- $r = H(\sum r_i) = H(g^k)$

Having `m`, the hash of the message we are signing (or a hash of the transaction), we start evaluating the signature with the following equation:

t-ECDSA Signature

Round 5

All players jointly decrypt w:
$\eta = TDec(w) = k\rho \bmod q$

and compute:
$\Psi = \eta^{-1} \bmod q = k^{-1}\ \rho^{-1}$

$$\begin{aligned}\sigma &= \Psi \times_e [(m \times_e u) +_e (r \times_e v)] \\ &= \Psi \times_e [E(m\rho) +_e E(r\rho x)] \\ &= (k^{-1}\ \rho^{-1}) \times_e [E(\rho(m + xr))] \\ &= E(k^{-1}\ (m + xr)) \\ &= E(s)\end{aligned}$$

After round 4, all players know those values:

- $u = \sum u_i = E(\rho)$ $\quad u_i \in Z_q$
- $v = \sum v_i = E(\rho x)$
- $k = \sum k_i$ $\quad k_i \in Z_q$
- $c = \sum c_i$ $\quad c_i \in [-q^6, q^6]$
- $w = \sum w_i = E(k\rho + cq)$
- $r = H(\sum r_i) = H(g^k)$

`c` is the value we have just evaluated, and `u`, `r` and `v` are the parameters jointly evaluated by all the signers in previous rounds.

So, since `u` is an encrypted `ρ`, and `v` is an encrypted `ρ` multiplied by the secret ECDSA key, we can do the following transformation:

t-ECDSA Signature

Round 5

All players jointly decrypt w:
$\eta = TDec(w) = k\rho \bmod q$

and compute:
$\Psi = \eta^{-1} \bmod q = k^{-1}\ \rho^{-1}$

$$\begin{aligned}\sigma &= \Psi \times_e [(m \times_e u) +_e (r \times_e v)] \\ &= \Psi \times_e [E(m\rho) +_e E(r\rho x)] \\ &= (k^{-1}\ \rho^{-1}) \times_e [E(\rho(m + xr))] \\ &= E(k^{-1}\ (m + xr)) \\ &= E(s)\end{aligned}$$

After round 4, all players know those values:

- $u = \sum u_i = E(\rho)$ $\quad u_i \in Z_q$
- $v = \sum v_i = E(\rho x)$
- $k = \sum k_i$ $\quad k_i \in Z_q$
- $c = \sum c_i$ $\quad c_i \in [-q^6, q^6]$
- $w = \sum w_i = E(k\rho + cq)$
- $r = H(\sum r_i) = H(g^k)$

Replace Ψ with the value it represents, we will get the following equation:

t-ECDSA Signature

Round 5

All players jointly decrypt w:
$\eta = TDec(w) = k\rho \bmod q$

and compute:
$\Psi = \eta^{-1} \bmod q = k^{-1}\ \rho^{-1}$

$$\begin{aligned}\sigma &= \Psi \times_e [(m \times_e u) +_e (r \times_e v)] \\ &= \Psi \times_e [E(m\rho) +_e E(r\rho x)] \\ &= (k^{-1}\ \rho^{-1}) \times_e [E(\rho(m + xr))] \\ &= E(k^{-1}\ (m + xr)) \\ &= E(s)\end{aligned}$$

After round 4, all players know those values:

- $u = \sum u_i = E(\rho)$ $\quad u_i \in Z_q$
- $v = \sum v_i = E(\rho x)$
- $k = \sum k_i$ $\quad k_i \in Z_q$
- $c = \sum c_i$ $\quad c_i \in [-q^6, q^6]$
- $w = \sum w_i = E(k\rho + cq)$
- $r = H(\sum r_i) = H(g^k)$

Eliminate ρ, we get this:

t-ECDSA Signature

Round 5

All players jointly decrypt w:
$\eta = TDec(w) = k\rho \bmod q$

and compute:
$\Psi = \eta^{-1} \bmod q = k^{-1}\ \rho^{-1}$

$$\begin{aligned}\sigma &= \Psi \times_e [(m \times_e u) +_e (r \times_e v)] \\ &= \Psi \times_e [E(m\rho) +_e E(r\rho x)] \\ &= (k^{-1}\ \rho^{-1}) \times_e [E(\rho(m + xr))] \\ &= E(k^{-1}\ (m + xr)) \\ &= E(s)\end{aligned}$$

After round 4, all players know those values:

- $u = \sum u_i = E(\rho)$ $\quad u_i \in Z_q$
- $v = \sum v_i = E(\rho x)$
- $k = \sum k_i$ $\quad k_i \in Z_q$
- $c = \sum c_i$ $\quad c_i \in [-q^6, q^6]$
- $w = \sum w_i = E(k\rho + cq)$
- $r = H(\sum r_i) = H(g^k)$

This is the equation is forthe standard ECDSA signature, where `k` is the cryptographically secure random integer, `n` is the message hash, `x` is our secret ECDSA key, and `r` is the curve generated point multiplied `k` times modulo `q`.

All those equations were done on ciphertexts, so at the end our signature is also encrypted:

t-ECDSA Signature

Round 5

All players jointly decrypt w:
$\eta = TDec(w) = k\rho \bmod q$

and compute:
$\Psi = \eta^{-1} \bmod q = k^{-1}\ \rho^{-1}$

$$\begin{aligned}\sigma &= \Psi \times_e [(m \times_e u) +_e (r \times_e v)] \\ &= \Psi \times_e [E(m\rho) +_e E(r\rho x)] \\ &= (k^{-1}\ \rho^{-1}) \times_e [E(\rho(m + xr))] \\ &= E(k^{-1}\ (m + xr)) \\ &= E(s)\end{aligned}$$

After round 4, all players know those values:

- $u = \sum u_i = E(\rho)$ $\quad u_i \in Z_q$
- $v = \sum v_i = E(\rho x)$
- $k = \sum k_i$ $\quad k_i \in Z_q$
- $c = \sum c_i$ $\quad c_i \in [-q^6, q^6]$
- $w = \sum w_i = E(k\rho + cq)$
- $r = H(\sum r_i) = H(g^k)$

Signature: Step 6 - Deal with encrypted results

All the players execute a threshold decryption mechanism to learn the value of `s`. And the decrypted value `s` and parameter `r` evaluated in round 4 together make the signature:

t-ECDSA Signature

Round 6

Players execute distributed decryption protocol TDec over the ciphertext $\sigma = E(s)$ to learn the value of s.

Players outputs (r, s) as the signature.

What is going on with Crypto Currencies

1. Lot's of borrowing in US
2. Interest rates go up - US gov owns lots of debt
3. Devalue the Currency
4. Move value into hard assets
5. Drive up value of hard assets