DSO:
Dependable Signing Overlay

Guofei Gu, Prahlad Fogla, **Wenke Lee**, Douglas Blough

Georgia Institute of Technology

Roadmap

- Motivation
- DSO Design: Architecture and Protocol
- Reliability and Security Analysis
- Conclusion

Motivation

- Digital signing is a very important component for computer and network security (to ensure integrity and provide authentication)
- A lot of applications, for example,
  - Public Key Infrastructure (PKI)
  - Certificate Authority (CA)
  - To distribute malware (worm/bot …) anti-body for quarantine

Dependable Service

- For a critical service such as digital signing, we require it to be *dependable*
- It should continue to provide service in case of
  - system failures (fault-tolerant)
  - malicious attacks (intrusion-tolerant)

Existing Work

- Redundancy
  - replicated servers
  - Byzantine quorum systems
- Problem: if one server is captured by an intruder, secret key can be stolen (confidentiality attack)

Existing Work (cont.)

- Threshold scheme *(k, m)*
  - secret sharing
  - threshold cryptography
- Problem
  - Confidentiality attack can succeed if *k* or more servers are compromised
  - Availability attack can succeed if *m-k+1* or more servers are under DoS attacks

Our Approach

- There are fixed critical lines (number of servers to attack) for an attacker to succeed in existing work
- Can we raise the bar?
- Yes, we can do better.
  - *Probabilistic critical line* (low probability for an attacker to succeed)

DSO: Dependable Signing Overlay

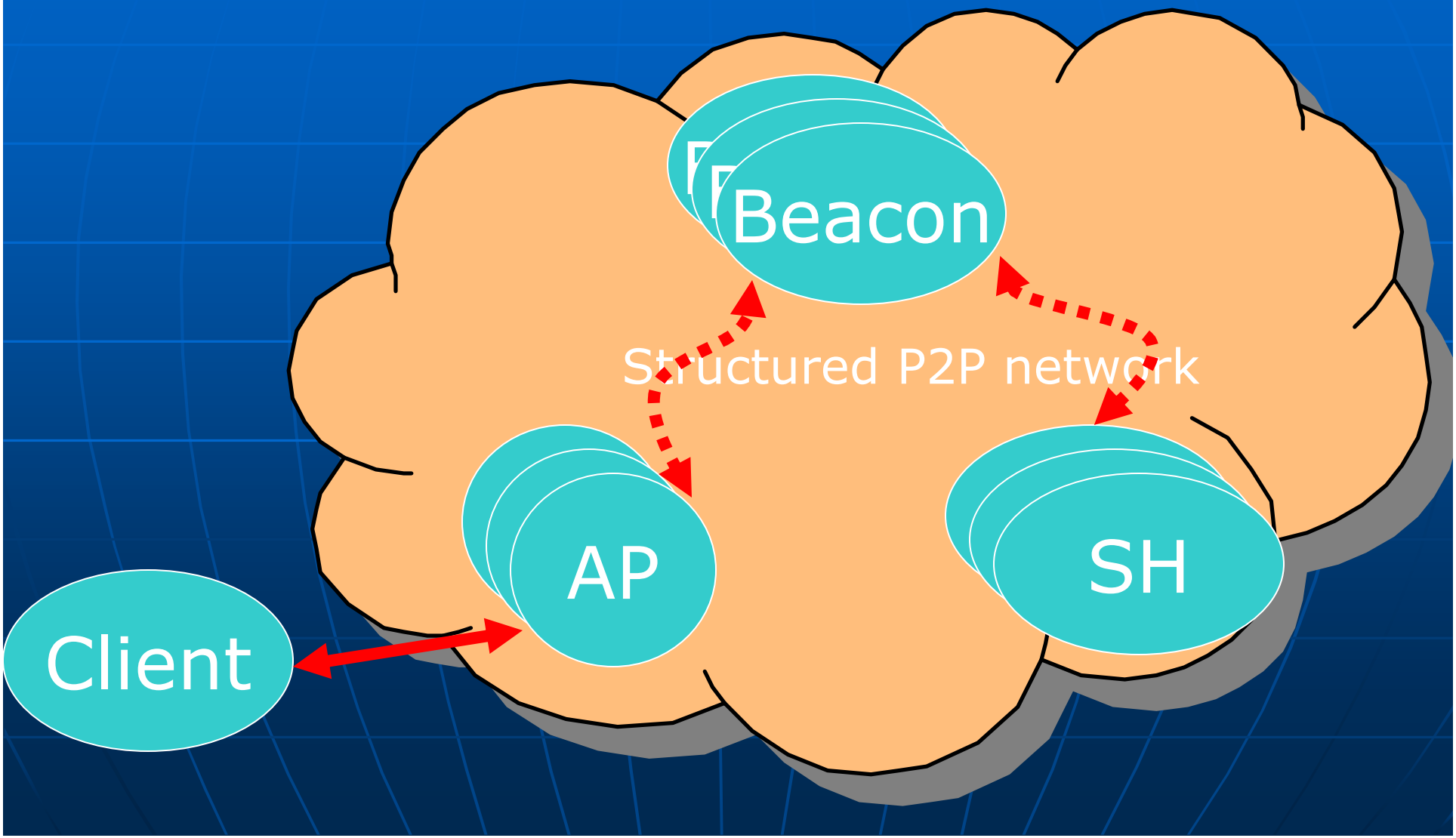
- Goal: dependable digital signing service with high fault-tolerance and high intrusion-tolerance

- Basic idea:
  - Hide the servers but still provide timely (bounded) service for legitimate clients
  - Threshold signing scheme + replicated servers

Structured P2P

- A typical structured P2P network based on DHT (distributed hash table) like Chord, CAN, PASTRY, Tapestry, …
  - Communication to any peer (ID) is guaranteed to succeed if and only if the peer exists
  - Communication to any peer (given ID) is guaranteed to terminate within a small and finite number of hops
  - The key ID space is uniformly divided among all currently active peers
  - The system is capable of handling dynamic peer joins and leaves

DSO Architecture



Service Initialization

- The service provider (initiator) generates m shares (Shamir's secret sharing)
- Randomly choose *m* nodes in DSO as SHs, distribute the shares
- SHs verify the shares (Feldman's verifiable secret sharing) and replicate each share to some other nodes
- All SHs notify all Beacons (identified with hash values of the service tag) about their roles and IDs.

Service Provision

- Client requests signing service (identified with the service tag) to any AP
- AP authenticates the client, and routes the request hop by hop to the destination Beacons *ID=Hash(tag)*
- Beacons forward request to SHs

Service Provision (cont.)

- SHs generate partial signed result (partial signature) and send to Beacon (combiner)
- Beacon receives *k* or more distinct partial signature, and verifies whether they are valid
- Beacon applies *K*-bounded coalition offsetting algorithm to obtain the final signature
- Final signature is sent to AP, then to the client
- Note: for privacy reason, the client can use blinding technique to hide the original message to sign.

Share Update

- The shares are periodically updated (proactive secret sharing) to enhance security

Robustness of DSO

- Multiple APs, beacons, and SHs for every service (redundancy)
  - Failure of one AP does not have much effect on the system. Client can simply choose another AP to enter the overlay.
  - If some beacon fails, DSO service can self-heal by choosing a new node as the beacon with using some new hash function.
  - If some SHs are compromised or targeted by attackers, the service provider can choose an alternate set of share holders and such operations are transparent to clients.
- Communication to some target node given its ID (not IP address) is indirect through a set of intermediate peer forwarding (anonymity)
  - All the beacons and SHs are anonymous to the clients (clients do not know the IP address of any SH or beacon for a service)

Fault-Tolerance Analysis

- Reliability of the whole system during *(0,t)* is the product of the following three items
  - Pr(At least one AP operates correctly during (*0,t*))
  - Pr(At least one beacon operates correctly during (*0,t*))
  - Pr(At least k distinct share holders operate correctly during (*0,t*))

Example

- Assume 10 different services, each using a (6,10) threshold scheme. Single reliability is 0.9.
  - The reliability of each service is 0.9984
- Now group these 10*10=100 nodes as a DSO, use the same (6,10) scheme with 10 beacons and 4 replication (for each distinct share). The same single reliability 0.9
  - The reliability of each service is 0.999999999 (nine 9's)

Example (cont.)

Single reliability	0.7	0.8	0.9	0.99
(6,10) scheme	0.8497	0.9672	0.9984	7 9's
DSO (n_b=10)	4 9's	6 9's	9 9's	15 9's
(60,100) scheme	0.9875	5 9's	15 9's	53 9's
DSO (n_b=15)	7 9's	10 9's	14 9's	15 9's

For DSO, total 100 nodes, 4 replication of each share, n_b is the number of beacons

Intrusion-Tolerance Analysis

- DSO is not to solve every security problems
  - we assume proper authentication, secure communication and routing, traffic analysis prevention and some intrusion detection techniques are used
- Threat model in our analysis
  - Availability attack: Attacker can launch DoS attacks. This may deny the signing service provided by the system to clients.
  - Confidentiality attack: Attacker can attack the nodes to obtain the shares and try to acquire the original service private key (signing key).
  - Integrity attack: Attacker can modify the shares on nodes in the overlay.

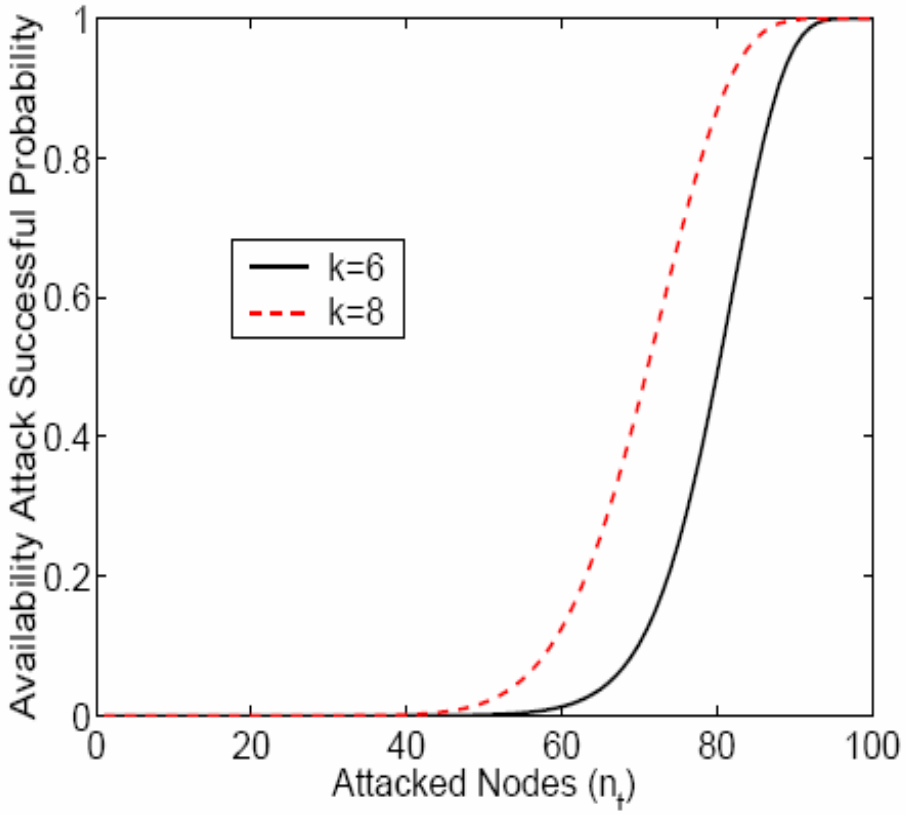
Availability Attack

- A given service will still be available if at least one AP is available, at least one beacon is available and *k* out of *m* distinct shares are available.

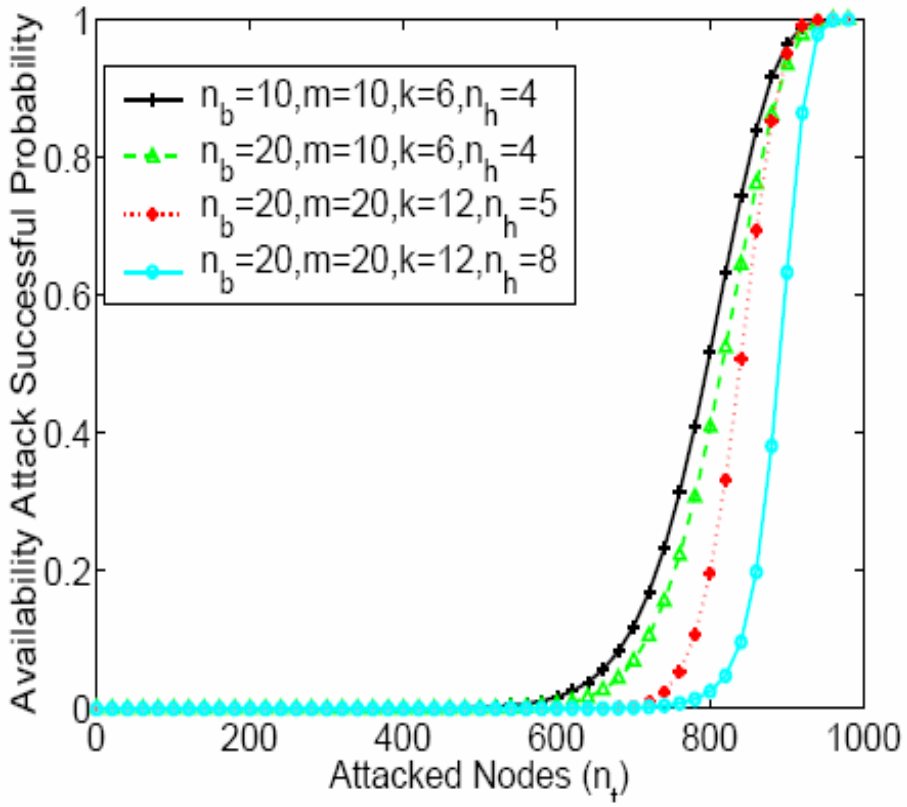
- We can compute the probability of successfully denying a given service
  - Using a basic block: $P_h(a,b,c)$ denotes the probability that randomly selected *b* nodes from totally *a* nodes, so that these *b* nodes contains a given set of *c* nodes

  - This probability is very low, e.g., using a (6,10) threshold scheme in a 100 node DSO, when 30 nodes are attacked, the probability of successfully compromising availability is only 0.0000034754.

Availability Attack: Examples



(a) Small DSO ($N_n = 100$, $n_a = 100, n_b = 10, m = 10, n_h = 4$)

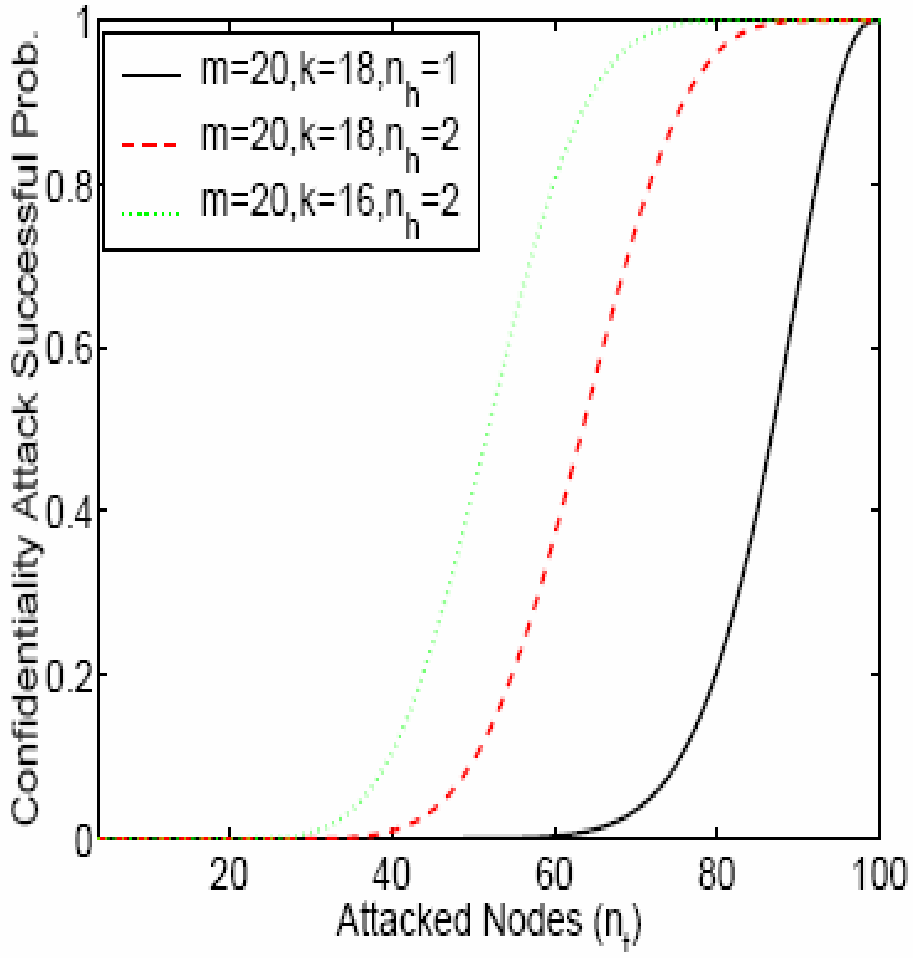


(b) Larger DSO ($N_n = 1,000, n_a = 1,000$)

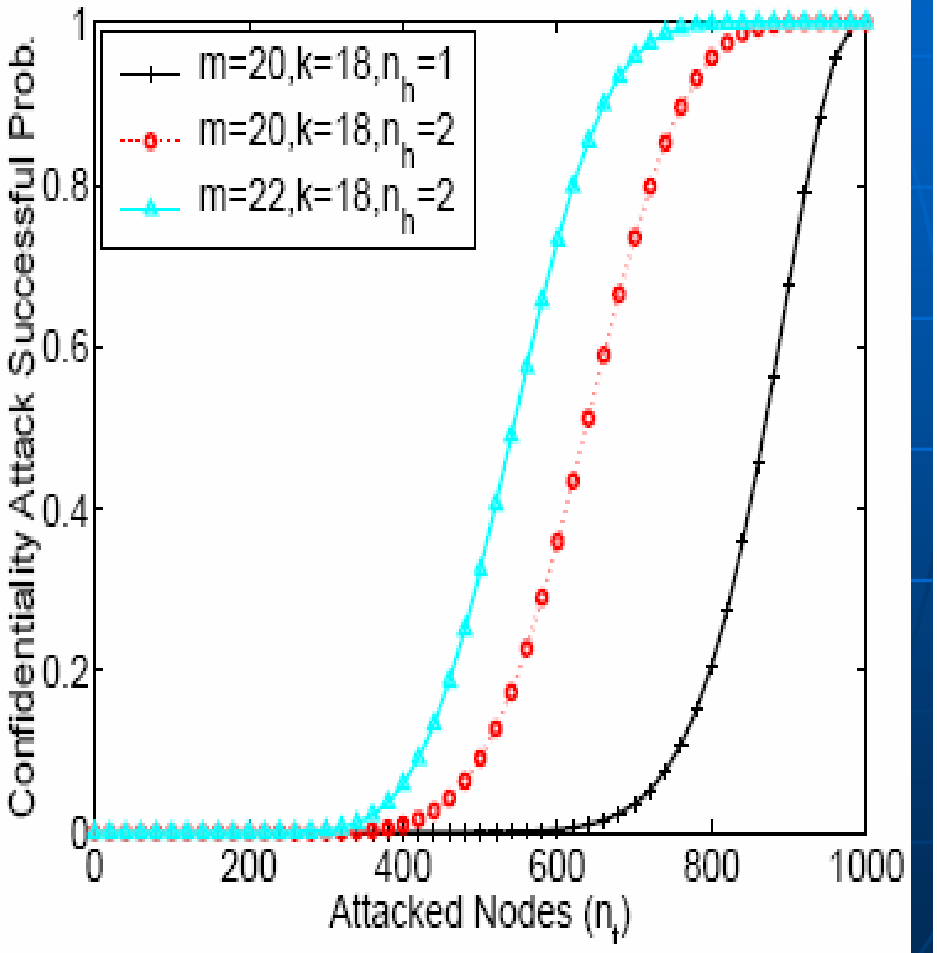
Confidentiality Attack

- A confidentiality attack may succeed only when an attacker can successfully get at least k distinct shares for a certain service.
- Basic analysis block: $P_g(a,b,c)$
  - the probability of randomly selecting *b* nodes from total *a* nodes, so that not contain any of the nodes from a given set of *c* nodes.
- Successful probability is also very low

Confidentiality Attack: Examples



(a) Small DSO ($N_n = 100$)



(b) Larger DSO ($N_n = 1,000$)

Integrity Attack

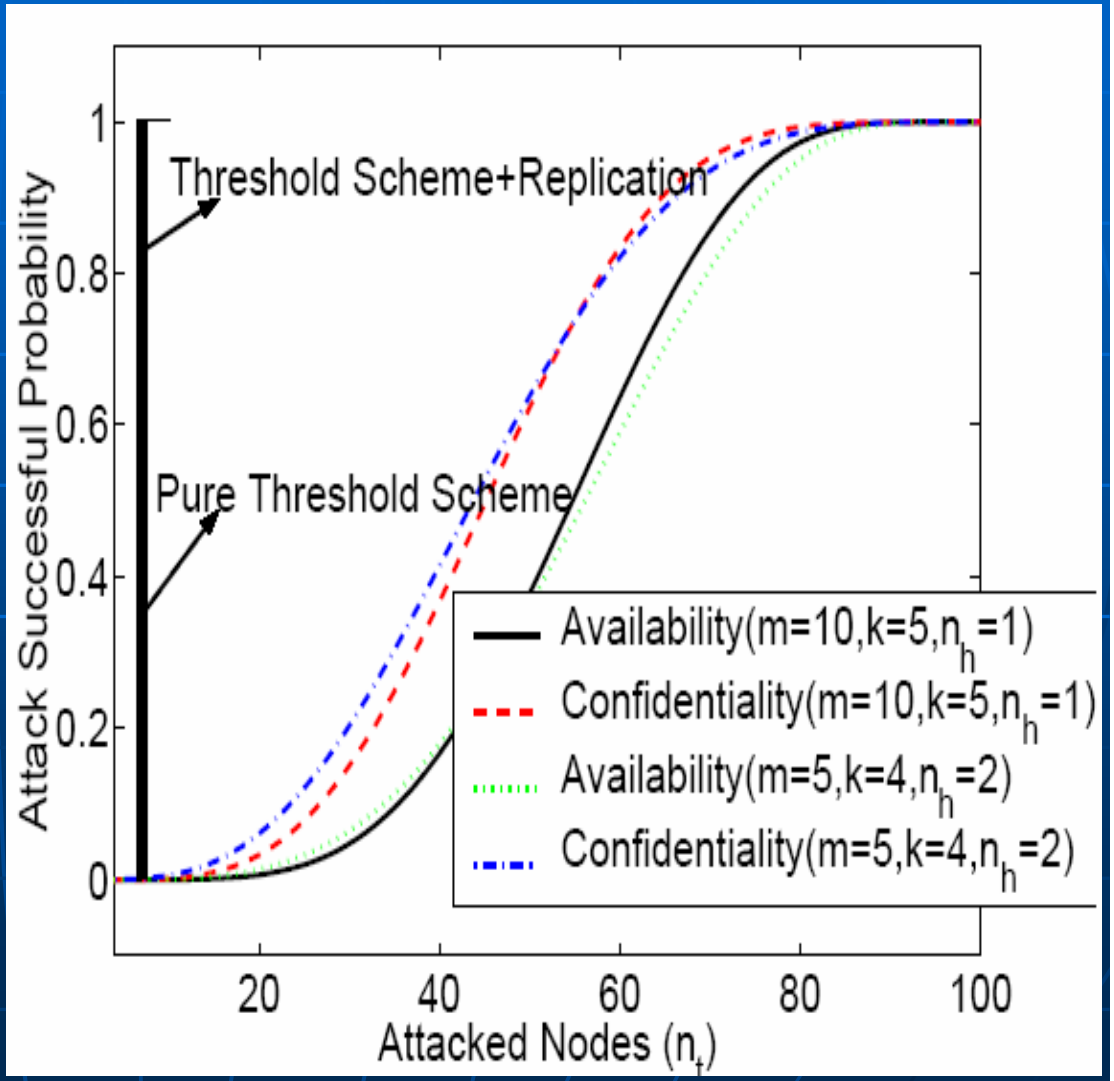
- Two aspects of an integrity attack
- Corrupt enough shares to forge a key
  - Analysis is similar to confidentiality attacks
- Modify enough distinct shares so that no clients are able to use the key
  - Analysis is similar to availability attacks

Discussion

- There is a trade-off between availability and confidentiality
  - Select proper parameters according to this trade-off
- Performance
  - Computation: use less nodes doing crypto operations, while provide similar security level as using a large number of nodes (doing crypto operations) in a threshold scheme
  - Communication: timely service with time bound *O(log(n))*

Comparison

- Provide a probabilistic (instead of fixed) critical line for an attacker to break
- The probability of being successfully attacked is very low in reasonable cases



Conclusion

- Anonymous servers using structured P2P (also providing timely service) + threshold scheme is a promising approach to build dependable systems
- We raise the bar for attackers to successfully attack a dependable system. The probabilistic (instead of fixed) critical line provides less chances for attackers
- DSO could be extended as a general, scalable architecture/infrastructure/platform to provide many other dependable services